Bootstrapping Techniques for Polysynthetic Morphological Analysis

William Lane
Charles Darwin University
william.lane@cdu.edu.au

Steven Bird
Charles Darwin University
steven.bird@cdu.edu.au

Abstract

Polysynthetic languages have exceptionally large and sparse vocabularies, thanks to the number of morpheme slots and combinations in a word. This complexity, together with a general scarcity of written data, poses a challenge to the development of natural language technologies. To address this challenge, we offer linguistically-informed approaches for bootstrapping a neural morphological analyzer, and demonstrate its application to Kunwinjku, a polysynthetic Australian language. We generate data from a finite state transducer to train an encoder-decoder model. We improve the model by "hallucinating" missing linguistic structure into the training data, and by resampling from a Zipf distribution to simulate a more natural distribution of morphemes. The best model accounts for all instances of reduplication in the test set and achieves an accuracy of 94.7% overall, a 10 percentage point improvement over the FST baseline. This process demonstrates the feasibility of bootstrapping a neural morph analyzer from minimal resources.

1 Introduction

Polysynthesis represents the high point of morphological complexity. For example, in Kunwinjku, a language of northern Australia (ISO gup), the word *ngarriwokyibidbidbuni* contains six morphs:

(1) ngarri- wok- yi- bid- bidbu- ni
1pl.excl- word- COM- REDUP- go.up- PI
'We were talking as we climbed up'

Example (1) illustrates common features of polysynthesis: fusion, incorporation, and reduplication. Fusion combines multiple grammatical functions into a single morph, leading to large morph classes, and challenging the item-and-arrangement leanings of finite state morphology. Incorporation presents a modelling challenge because rule-based methods are unable to enumerate an open class, and machine learning methods need to learn how to recognize the boundary between contiguous large or open morph classes. Reduplication is also a challenge because it copies and prepends a portion of the verb root to itself, requiring a nonlinear or multi-step process. Tackling these phenomena using finite state transducers (FSTs) involves a combination of technical devices whose details depend on subtleties of the morphological analysis (cf. Arppe et al., 2017). There remains a need for more investigation of polysynthetic languages to deepen our understanding of the interplay between the options on the computational side, and the most parsimonious treatment on the linguistic side.

Morphological complexity leads to data sparsity, as the combinatorial possibilities multiply with each morpheme slot: most morphologically complex words will be rare. Furthermore, many morphologically complex languages are also endangered, making it difficult to collect large corpora. Thus, polysynthetic languages challenge existing ways of building tools and applications for the communities that speak these languages.

In this work we investigate Kunwinjku, spoken by about 2,000 people in West Arnhem in the far north of Australia. Members of the community have expressed interest in using technology to support language learning and literacy development. Thus, we face the challenge of developing useful language technologies on top of robust models, with few resources and in a short space of time. We envisage morphologically-aware technologies including dictionary interfaces, spell checkers, text autocompletion, and tools for language learning (cf. Littell et al., 2018).

This paper is organized as follows. We begin by reviewing previous work in finite state morphology,

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6652–6661
July 5 - 10, 2020.

low resource morph analysis, neural approaches to morph analysis, and data augmentation for morphological reinflection (Sec. 2). Next, we describe our existing finite state model for Kunwinjku verbs (Sec. 3). In Section 4 we present a neural approach which addresses gaps in the previous model, including the ability to analyze reduplication and to exploit distributional information. Next we discuss our evaluation metrics and our handling of syncretism and ambiguity (Sec. 5). Finally, the results are presented in Section 6, including a discussion of how well the neural models address the shortcomings of the FST model.

Our contributions include: (a) a robust morphological analyzer for verbs in a polysynthetic language; (b) a method for augmenting the training data with complex, missing structure; and (c) a technique for scoring the likelihood of generated training examples.

2 Background and Related Work

Finite state transducers (FSTs) are a popular choice for modelling the morphology of polysynthetic languages. Several toolkits exist, including XFST, Foma, and HFST (Beesley and Karttunen, 2003; Hulden, 2009; Lindén et al., 2013). Each one is an optimized implementation of the finite state calculus (Kaplan and Kay, 1994), providing additional support for morphosyntactic and morphophonological processes. Most recent work on computational modelling of morphologically rich languages is built on the foundation of these tools (Arppe et al., 2017; Littell, 2018; Andriyanets and Tyers, 2018; Chen and Schwartz, 2018; Cardenas and Zeman, 2018). As a case in point, we applied Foma in the analysis of the morphology of Kunwinjku verbs, but ran into difficulties accounting for out-of-vocabulary (OOV) items in open morph classes. We also stopped short of addressing complex features like reduplication and verbal compounding, for technical reasons related to the expressiveness of FSTs (cf. Lane and Bird, 2019).

Recently, neural models have gained popularity for morphological processing because they address some of the weakness of FSTs: subword modeling shows an ability to remain robust in the face of out-of-vocabulary items, and recurrent neural architectures with attention have shown a capacity to learn representations of context which allow the model to incorporate the notion of long-distance dependencies (Bahdanau et al., 2014).

Neural morphological analyzers can be developed from training data generated by an FST. These analyzers are more robust, handling variation, out-of-vocabulary morphs, and unseen tag combinations (Micher, 2017; Moeller et al., 2018; Schwartz et al., 2019). They provide 100% coverage, always providing a "best guess" analysis for any surface form. Of course, FSTs can be modified to accommodate exceptions and OOV morphs, but this requires explicit modelling and usually does not achieve the robustness of neural analyzers (Schwartz et al., 2019).

Anastasopoulos and Neubig (2019) found that they could augment their training set by hallucinating new stems, increasing accuracy on their test set by 10 percent. This method involved substituting random characters from the target language's alphabet into the region identified by alignment as the probable root. For the sake of cross-lingual generalizability, their method does not consider language-specific structure.

The task of morphological analysis, mapping an inflected form to its root and grammatical specifications, is similar to the task of machine transliteration, mapping a sequence of words or characters from source to target language without reordering. For example in Kunwinjku, consider the segmentation and gloss of the verb *karridjalbebbehni*:

(2) karri- djal- bebbeh- ni
12a- just- DISTR- sit.NP
'Let's just sit down separately' [E.497]

Since the process of segmenting and glossing the verb does not contain any reorderings, the mapping of surface to glossed forms can be viewed as transliteration.

3 A Finite State Model of Kunwinjku

Finite state transducers have long been viewed as an ideal framework to model morphology (Beesley and Karttunen, 2003). They are still a popular choice for low-resource polysynthetic languages (cf. Chen and Schwartz, 2018; Lachler et al., 2018). Here we summarize some features of Kunwinjku and describe the finite state implementation.

3.1 Features of Kunwinjku

Kunwinjku is a polysynthetic agglutinating language, with verbs having up to 15 affix slots (Fig. 1). Morphs combine in a way that is "almost lego-like" (Evans, 2003; Baker and Harvey, 2003).

−12	−11	−10	(−9)	(−8)	(−7)	(−6)	(−5)	(−4)	(−3)	(−2)	(−1)	0	+1	+2
Tense	Subject	Object	Directional	Aspect	Misc1	Benefactive	Misc2	GIN	BPIN	NumeroSpatial	Comitative	Verb root	RR	TAM

Figure 1: Verbal affix positions in Kunwinjku. Regions where indices share a cell ([−12,−10], [+1,+2]) indicate potentially fused segments. Slot indices in parentheses indicate optionality. Adapted from (Evans, 2003, Fig 8.1).

We implement morphotactics and morphophonology as separate stages, following usual practice (Fig. 2). However, this is not conducive to modelling noun incorporation, valence-altering morphology, fusion, or reduplication, all typical phenomena in polysynthetic languages.

Kunwinjku has two kinds of noun incorporation. General incorporable nouns (GIN) are a closed class, manifesting a variety of grammatical roles (3). Body part incorporable nouns (BPIN) are an open class, restricting the scope of the action (4).

(3) nga- kak- keleminj
1m- night- fear.P
'I was afraid at night'

(4) nga- bid- keleminj
1m- hand- fear.P
'I was afraid for my hand' [E.458]

The open class BPIN occupy slot −3 and will be adjacent to the verb root whenever slots −2 and −1 are empty, as is common. With adjacent open class slots, Kunwinjku opens up the possibility of there being *contiguous OOV morphs*. In Kunwinjku there is no template to help distinguish members of these adjacent classes, thus creating a novel challenge for predicting morph boundaries.

While transitivity of the verb is lexically defined, there are three morph classes which signal valency change: the benefactive (BEN), comitative (COM), and reflexive (RR). More details about the respective function of these morphs is given in Lane and Bird (2019), but here it suffices to say their presence in a verb makes resolving valency impossible without wider sentential context. This impacts the FST modelling, as we are unable to restrict possible illegal analyses on this basis, which results in overgeneration.

Morphological fusion can lead to a proliferation of morphs and analyses. In Kunwinjku, there are no fewer than 157 possibilities for the first slot of the verb, fusing person and number (for both subject and object) along with tense. We find that this fusion affects decisions around tokenization of the data in preparation for training the seq2seq model (Sec. 4.2).

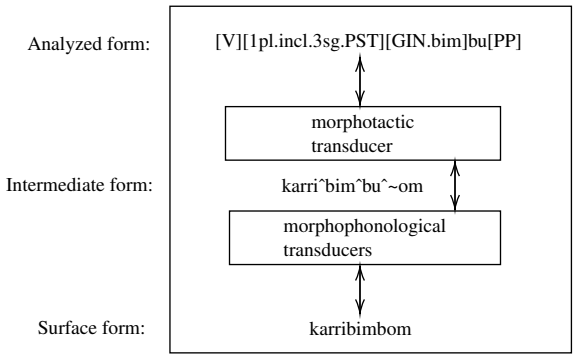

Figure 2: The high-level structure of the Kunwinjku finite state transducer. Analyzed forms are mapped to surface forms (and vice versa) through the composition of morphotactic and morphophonological transducers.

Most of the world's languages employ reduplication productively for diverse purposes (Rubino, 2005). It is a common feature of polysynthetic languages in particular. While modelling reduplication using FSTs is possible, the general consensus is that modelling partially reduplicative processes explode the state space of the model, and are burdensome to develop (Culy, 1985; Roark et al., 2007; Dras et al., 2012). For these reasons, the Kunwinjku FST model does not include an implementation of the language's complex reduplication system.

In Kunwinjku, there are three types of verbal reduplication: iterative, inceptive, and extended. Each type of reduplication has 1–3 (CV) templates which can be applied to the verb root to express the semantics associated with each type. In Section 4.4 we discuss an approach to ensure that the neural model handles Kunwinjku's complex reduplication system.

3.2 Evaluating the FST

We establish a baseline by scoring the FST on a set of $n = 304$ inflected verbs. The data was collected from the Kunwinjku Bible (which targets a modern vernacular), a language primer (Etherington and Etherington, 1998), and a website (Bininj Kunwok Language Project, 2019). The data was glossed in consultation with language experts.

We define coverage as number of analysed forms, and accuracy as the number of *correctly* analyzed forms, both as a fraction of n. We define precision

	Accuracy	Coverage	Precision
FST	84.4	88.5	95.4

Figure 3: All-or-nothing accuracy and coverage of the Kunwinjku FST Analyzer on the test set of 304 inflected verbs.

Error Class	% of Error
Reduplication	28.9
TAM Inflection	28.5
OOV root	26.3
OOV inc. nominals	13.2
Alternation	2.2

Figure 4: Error analysis of Lane and Bird (2019)'s FST model of Kunwinjku verbs shows 5 classes of error and the percent of the total error attributed to each class.

as the number of correctly analysed forms as a fraction of the number of analysed forms. We distinguish accuracy and precision because the ability of a model to withhold prediction in case of uncertainty is useful in certain application contexts.

The results of the evaluation show that while the FST is fairly high-precision, its accuracy is limited by the imperfect coverage of verb stems in the lexicon (Fig. 3).

The FST relies on a lexicon to provide analyses for inflected forms, and when it comes across OOV morphs, or verb stems modified by processes like reduplication, it fails to return an analysis. We sort the coverage issues into classes, and remark that the largest source of error comes from reduplication, followed by variation in tense/aspect/mood (TAM) inflection, OOV stems, OOV incorporated nominals, and exceptions to the d-flapping alternation rule (Fig. 4). We address each of these problems in the following sections.

4 Methods

In this section we discuss the approach which leverages an incomplete FST to produce a more robust neural morphological analyzer for Kunwinjku. Those steps include generating training pairs from an FST, tokenizing the data, resampling from the dataset to simulate distributional signal, hallucinating missing structures into the dataset, and training a neural encoder-decoder model on the resampled data.

4.1 Data generation from an FST

Given our low resource setting, training a neural encoder-decoder model like those used in neural machine translation (NMT) is not possible without augmenting what resources we do have. Following the established template of recent work on neural morphological analysis for low resource polysynthetic languages (Micher, 2017; Moeller et al., 2018; Schwartz et al., 2019) we use the FST model to generate morphotactically valid pairs of surface and analyzed verbs.

For the purpose of training the base neural model, we adapted the Foma tool to randomly generate 3,000,000 surface/analysis pairs from the FST (see Fig. 6 for an example of a tokenized pair). An automatic process removed duplicates, leaving us with 2,666,243 unique pairs which we partitioned into an .8/.1/.1 train/dev/test split.

In Schwartz et al. (2019)'s work on modelling complex nouns in Yupik, they generate a training set which exhaustively pairs every Yupik noun root with every inflectional suffix, regardless of the resulting semantic fidelity. In our case, it was not feasible to exhaustively generate the training data, as it would have led to 4.9×10^{12} instances (Fig. 5). In effect, the training set represents .00004% of the space over which we seek to generalize.

4.2 Tokenization

To prepare the data for training a seq2seq model, we first collect the glossed inflected verb forms, perform tokenization, and organize them into source-target pairs.

We chose a tokenization scheme which treats graphemes as atomic units. Morph labels are also treated mostly as atomic units, with the exception being for fused labels which we break into their individual linguistic components (Fig. 6). For example the pronominal morph in Kunwinjku can simultaneously express both subject and object, as well as tense. Consider the pronominal prefix *kabenbene-* which we gloss as 3sg.3ua.nonpast and tokenize as [3sg . 3ua . nonpast]. Choosing to break up labels in the fused morphological slots prevents an unnecessary proliferation of entries in the target vocabulary, as individual units like 3sg, 3ua, and past can be shared by multiple pronominals. Our choice to tokenize the source forms and verb root strings at the grapheme level reflects our desire to loosen the model's vocabulary such that it is

TSO		DIR		ASP		MSC1		BEN		MSC2		GIN		BPIN		COM		root		RR		TAM		*Total*
157	x	3	x	2	x	24	x	2	x	4	x	78	x	32	x	2	x	541	x	2	x	5	=	$4.9x10^{12}$

Figure 5: An estimate for all morphotactically valid sequences covered by the Kunwinjku FST

equipped to handle variation at the orthographic level, and possible OOV stems.

4.3 Simulating distributional information

Generating from an FST at random fails to capture valuable information about the distribution of morphs. For example in Kunwinjku, body part incorporable nouns (BPIN) can occur adjacent to the verb root. Both categories are open class, meaning that there is a high likelihood in the low-resource setting that either or both are out-of-vocabulary. How then does the analyzer decide where to place the boundary? Perhaps the entire sequence is a single out-of-vocabulary root. Our intuition is that knowing the likelihood of co-occurrence for two analysis tags can provide signal to help disambiguate. Some morph sequences are inevitably more frequent than others, and we would like to represent that information in the training set.

To this end, we propose a method for simulating distributional information in the training set. First, we want to score any analyzed form, giving higher scores to forms that contain more likely sequences. We define M as the sequence of morph tags which make up an analysis, where m_i is the morph tag at index i. The scoring function is defined as follows:

$$\text{score}(M) = \frac{1}{n}\sum_{i}^{n} \log P(m_i, m_{i+1}) \quad (5)$$

The joint probability of adjacent tags is estimated from a corpus of unannotated text, here, selected books from the Kunwinjku Bible. Everything the existing FST can analyse as a verb is considered to be a verb, and is used to calculate the joint probability table.

The training set is tagged with the FST[1], and ranked according to the scoring function. We split the sorted data into buckets defined by their morphotactic likelihood, and then sample from them according to a Zipf distribution. The effect is that more probable sequences are more likely to occur in the training data than less likely examples, thus approximating the distribution of morphotactic structure we would expect to see in a natural corpus.

[1]By using an FST with imperfect recall we are not capturing true distributional information; it is simply a heuristic.

4.4 Hallucinating reduplicative structure

One shortcoming of the Kunwinjku FST model is that it does not account for reduplicative structure, due to the complexity of modelling recursive structure in the linear context of finite state machines (Culy, 1985; Roark et al., 2007). As noted previously, reduplication is responsible for 28.9% of the FST's coverage error when evaluated on the test set of inflected verbs. If reduplication is not modeled by the FST, then reduplication will also not be represented in the training set generated by that FST. We posit that if data hallucination has been shown to improve performance in the language-agnostic setting (Anastasopoulos and Neubig, 2019; Silfverberg et al., 2017), than it is likely that linguistically-informed hallucination can provide a similar reinforcement in Kunwinjku. In line with this, we developed an extension to the data generation process which hallucinates reduplicative structure into a subset of the training data.

Kunwinjku has three main types of partial verbal reduplication signaling iterative, inceptive, and extended meaning. Moreover, each type of reduplication can have more than one CV template, depending on which paradigm the verb belongs to. Figure 7 documents the three types of reduplication, and serves as the template for the reduplicative structure hallucinator.

First, the hallucinator module samples n% of the FST-generated pairs and strips away the affixes to isolate the root. For each root, one of the three reduplication types (iterative, inceptive, or extended) is selected at random, and the root is matched against the available CV templates. The longest pattern which matches the root is selected, and the pattern-matching portion of the root is copied and prepended to the root. Both the surface and analyzed form are updated to reflect the change, and the new training pairs are appended to the original list of FST-generated pairs.

4.5 Training

We trained an encoder-decoder model on the dataset of 2,114,710 surface/analyzed form pairs (the Base model). We then hallucinate reduplication into 8% of the Base data, and

```
b i k a nj ng u n e ng -> [ 3sg . 3Hsg . PST] [ BPIN ] ng u [ PP ]
```

Figure 6: An example of a tokenized source/target training pair, where we treat source graphemes, target labels, fused target label components, and verb root graphemes as atomic units.

Type	Pattern(s)	Unreduplicated Verb	Reduplicated Verb	Semantic Effect on Verb (V)
Iterative	CVC	dadjke = cut	dadj-dadjke = cut to pieces	Doing V over and over again
	CV(C)CV(h)	bongu = drink	bongu-bongu = keep drinking	
	CVnV(h)	re = go	rengeh-re = go repeatedly	
Inceptive	CV(n)(h)	yame = spear (sth)	yah-yame = try (and fail) to spear (sth)	Failed attempt to do V
		durnde = return	durnh-durnde = start returning	Starting to do V
Extended	CVC(C) || _ men	djordmen = grow	djordoh-djordmen = grow all over the place	Doing V all over the place
	CVC(C) || _ me	wirrkme = scratch	wirri-wirrkme = scratch all over	

Figure 7: Reduplication in Kunwinjku has three forms, and each form has its own CV templates defining how much of the verb is captured and copied. In the case where we've used the form X || _ Y, we mean that pattern X is the reduplicated segment if found in the context of Y. Figure adapted from (Evans, 2003).

combine that hallucinated data to the base training data set (the Base+halluc[...] models).

The model setup is similar to the one described in (Schwartz et al., 2019). We use MarianNMT: a fast, open-source toolkit which implements neural models for machine translation (Junczys-Dowmunt et al., 2018). We used a shallow attentional encoder-decoder model (Bahdanau et al., 2014) using the parameters described in (Sennrich et al., 2016): the encoder and decoder each have 1 hidden layer of size 1024. We use cross-validation as the validation metric, set dropout to .2 on all RNN inputs, and enable early stopping to avoid overfitting. We use the same setup and parameters for all NMT models mentioned in this paper. A full accounting of the MarianNMT settings used can be seen in the Appendix.

5 Evaluation of the Neural Models

We begin by reporting the performance of the neural models in terms of coverage, accuracy, and precision, so that they can be compared with the evaluation of the FST model, described in Section 3.2. Additionally, we measure the performance of the neural models in terms of precision (P), recall (R), and F1 on the morph level: For each morph tag in the gold target test set, we calculate P, R, and F1, and then calculate the macro-average P, R, and F1 across all tags in the test set (Fig. 9). This method is more granular than all-or-nothing accuracy over the entire translated sequence, and allows us to get a better picture of how the models are doing on the basis of individual tags.

We observed an issue with syncretic ambiguity which complicates the evaluation process (also noted by Schwartz et al. 2019; Moeller et al. 2018). For example, the pronominal prefix *kabindi-* can be glossed: [3ua.3ua.nonpast], or [3pl.3ua.nonpast], or [3ua.3pl.nonpast], or [3pl.3pl.nonpast]. Here, the pronominal expresses both the subject and object, and is not explicit whether that subject or object is the 3rd person dual or plural, in any of four possible combinations. The disambiguation cannot be resolved at the level of the isolated verb.

Our initial experiment with the base data set achieved 100% coverage and 68.3% accuracy on the test set. When confronted by the same problem, Moeller et al. (2018) decided to collapse ambiguous tags into an underspecified meta-tag. For example, for the Kunwinjku data, we might collapse the four tags above into [3pl.3pl.nonpast]. However, doing so results in a potential loss of information. Given the wider sentential context, the pronominal could be possibly be disambiguated, so long as the distinction is preserved and all equally-valid analyses are returned.

Further, as Schwartz et al. (2019) point out, in the Yupik language it is possible for this ambiguity to exist across other categories which are not easily collapsed. In Kunwinjku, an example of this would be the pronominals [1sg.2.past] and [3sg.past] which differ in terms of number and valency, and yet share the same null surface form. Their differences are such that they can not be easily collapsed into a single meta-tag. Therefore we do not penalize the model for producing any variation of equally valid analyses given the surface form, and for each model we adjust the evaluation for syncretism in a post-processing step.

6 Results and Discussion

All of the neural models outperform the FST in terms of accuracy and coverage (Fig. 8). However, the FST is more precise, and this may be useful in certain application contexts. The best model is Base+halluc+resample, which improves on the FST by 10.3 percentage points. On the morph-level, we see that the neural models containing the hallucinated reduplication data outperform the base neural model (Fig. 9).

	Acc	Cov	Precision
FST	84.4	88.5	**95.4**
Base	89.1	**100**	89.1
Base+halluc	93.7	**100**	93.7
Base+halluc+resample	**94.7**	**100**	94.7

Figure 8: All-or-nothing accuracy and coverage of the three morphological analyzer models

	Precision	Recall	F1
Base	88.8	89.9	89.0
Base+halluc	91.6	92.6	91.8
Base+halluc+resample	**93.7**	**93.6**	**93.4**

Figure 9: Morph-level performance of shallow neural sequence models. Macro P/R/F1 across all morph tags.

We posited that the difficulties encountered by the FST model—namely reduplication, out-of-vocabulary items, and spelling variation—could be at least partially addressed by training a neural model on character and tag sequences, and hallucinating instances of reduplication into the training set. For the most part, this held true, as we see gains across all error classes (cf. Sec. 3.2). Here we report performance with respect to the three largest error classes: reduplication, OOV verbs, and OOV nouns.

6.1 Reduplication

As expected, neither the FST nor the Base neural model succeeds in recognizing reduplication. It would be impossible, as the REDUP tag does not appear in either of their vocabularies.

The Base+halluc model's performance gain over the Base model can be accounted for entirely by the fact that it achieved 100% recall of reduplicative structure. Precision, on the other hand was 57.9%. Looking at the errors, we find that the imprecise predictions were all applied to instances about which the system was already wrong in previous models, meaning that the impact of reduplicative hallucination between models was only positive. In the Base+halluc+resample model, recall of reduplicative structure was also 100%, and precision increased slightly to 58.8%.

Unseen Verbs	**Base+halluc+resample**	✓/✗
wobekkang	[GIN]bekka	✗
nga**kohbanjm**inj	[GIN][REDUP]me	✗
nga**rrukkendi**	dukkendi	✓
ka**menyime**	[GIN]yime	✗
yimalng**darrkiddi**	darrke[PERSIST]	✗
ngam**dolkka**ng	[DIR][GIN]ka	✗
dolkkang	[GIN]ka	✗
ka**rrukmirri**	dukmirri	✓
ngurrimirnde**mornname**rren	mornname	✓

Unseen GIN/BPIN/ASP	**Base+halluc+resample**	✓/✗
kan**njilng**marnbom	[GIN]	✗
yiben**kange**marnbom	[REDUP]	✗
kan**kange**murrngrayekwong	[GIN]	✗
kankange**murrng**rayekwong	[BPIN]	✓
kankangemurrng**rayek**wong	[REDUP]	✗
kan**kange**marnbom	[REDUP]	✗
ngarri**bangme**marnbuyi	[BPIN]	✗
yi**malng**darrkiddi	[GIN][REDUP]	✗

Figure 10: Column 1 shows the list of verbs and nouns (in bold) which are are unseen in the FST lexicon. Column 2 is the Base neural model's prediction covering the character sequence corresponding to the unseen item. Column 3 indicates whether the neural model's analysis of the morph is correct.

6.2 Discovering New Lexical Items

The neural models correctly identify some unseen verb stems, but still show room for improvement. We observe a tendency across all neural models to predict verb stems which have been seen in training, and which are also a substring of the observed unknown root. For example, the training set does not contain any verbs with the root *dolkka*, but it shows up 3 times in the test set. The analyses of all *dolkka*-rooted verbs were the same in both the Base+halluc and Base+halluc+resample models: they propose *ka*, a known root from the training set, and presume *dolk-* to be an incorporable noun[2]. Figure 10 shows a sample of OOV verb stems and nouns from the test set. In the unseen verbs table, this behavior of preferring previously observed verb stems is the cause of error in every case.

Further difficulty comes in distinguishing between general (GIN) and body-part (BPIN) incorporated noun classes. The low rate of success in positing unknown incorporated nouns is, in

[2]Possibly by virtue of its orthographic proximity to *bolk-*, a common general incorporable noun which means "land."

large part, attributed to the fact that the large GIN and open BPIN classes often occur adjacent to each other and to the root. The neural model has difficulty making useful predictions when multiple morphs in this region are previously unobserved.

Overall, the Base+halluc+resample model correctly posited 33% of unseen stems, and 12.5% of unseen nouns from the FST error analyses.

6.3 Impact of distributional information

This technique to approximate distributional information led to a small improvement in overall accuracy, and in tag-level P/R/F1. We had expected that this information might help the neural models learn something about the relative frequencies of GINs or BPINs, which could help make decisions about how to draw the boundary between unseen stems and unseen incorporated nominals. Instead, we saw distributive information helped to disambiguate the boundaries between morph classes with fewer members.

One representative example is the case of *yikimang*, whose root is *kimang*. Before resample, the neural models interpret the *yi-* as the comitative prefix *yi-*, and injects a spurious COM tag into the analysis. After resample, it correctly omits the COM tag, interpreting *yi-* as the 2nd person singular pronominal. In the unfiltered FST-generated training data, COM occurs in 53% of instances. In the resampled data, it occurs in 22% of instances. When all morph labels are equally likely to occur, the model is just as likely to predict any morph label compatible with the character sequence. Resampling the training data according to a more realistic distribution leads to stronger morph transition priors, which tip the scale in favor of the analysis with a more likely tag sequence.

7 Conclusion

We have shown that complex features of polysynthetic morphology, such as reduplication and distributional morphotactic information, can be simulated in the dataset and used to train a robust neural morphological analyzer for a polysynthetic language. In particular, we showed that a robust neural model can be bootstrapped in a relatively short space of time from an incomplete FST.

This work represents a successful first iteration of a process whereby the morphological model can be continually improved. Indeed, the concept of *bootstrapping* a model implies an iterative development story where much of the scaffolding used in early efforts will eventually fall away. For example, once the bootstrapped model has been used to tag verbs containing reduplication, we can confirm the model's high-confidence predictions and retrain. In this second iteration, we may find that we no longer need to hallucinate reduplication because it is sufficiently represented in the new training set. Similarly, once we have applied the complete neural model to a corpus of natural text, we will no longer need to approximate distributional information. For researchers developing robust morphological analyzers for low resource, morphologically complex languages, this work represents a template of model development which is well-suited for the context.

Producing a viable morphological analyzer is the first step towards building improved dictionary search interfaces, spell-checking tools, and computer-assisted language learning applications for communities who speak low-resource languages. The pattern of training robust systems on data that has been augmented by the knowledge captured in symbolic systems could be applied to areas outside of morphological analysis, and is a promising avenue of future exploration.

Acknowledgments

We are grateful for the support of the Warddeken Rangers of West Arnhem. This work was covered by a research permit from the Northern Land Council, and was sponsored by the Australian government through a PhD scholarship, and grants from the Australian Research Council and the Indigenous Language and Arts Program. We are grateful to four anonymous reviewers for their feedback on an earlier version of this paper.

References

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 984–996, Hong Kong, China. Association for Computational Linguistics.

Vasilisa Andriyanets and Francis Tyers. 2018. A prototype finite-state morphological analyser for Chukchi. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 31–40, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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 Athabaskan (Dene) Languages*, pages 51–69.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Brett Baker and Mark Harvey. 2003. Word Structure in Australian Languages. *Australian Journal of Linguistics*, 23:3–33.

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

Bininj Kunwok Language Project. 2019. Bininj Kunwok: kunwok dja mankarre kadberre–our language, our culture. https://bininjkunwok.org.au/. Accessed: 2019-10-10.

Ronald Cardenas and Daniel Zeman. 2018. A morphological analyzer for shipibo-konibo. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 131–139.

Emily Chen and Lane Schwartz. 2018. A morphological analyzer for St. Lawrence Island/Central Siberian Yupik. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

Christopher Culy. 1985. The complexity of the vocabulary of Bambara. In *The Formal Complexity of Natural Language*, pages 349–357. Springer.

Mark Dras, François Lareau, Benjamin Börschinger, Robert Dale, Yasaman Motazedi, Owen C Rambow, Myfany Turpin, and Morgan Elizabeth Ulinski. 2012. Complex predicates in arrernte.

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.

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.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1804.00344*.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Jordan Lachler, Lene Antonsen, Trond Trosterud, Sjur Moshagen, and Antti Arppe. 2018. Modeling Northern Haida verb morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

William Lane and Steven Bird. 2019. Towards A Robust Morphological Analyzer for Kunwinjku. In *Proceedings of ALTA 2019*. None yet.

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.

Patrick Littell. 2018. Finite-state morphology for kwak'wala: A phonological approach. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 21–30, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Patrick Littell, Anna Kazantseva, Roland Kuhn, Aidan Pine, Antti Arppe, Christopher Cox, and Marie-Odile Junker. 2018. Indigenous language technologies in canada: Assessment, challenges, and successes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2620–2632.

Jeffrey Micher. 2017. Improving coverage of an inuktitut morphological analyzer using a segmental recurrent neural network. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 101–106.

Sarah Moeller, Ghazaleh Kazeminejad, Andrew Cowell, and Mans Hulden. 2018. A neural morphological analyzer for arapaho verbs learned from a finite state transducer. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 12–20.

Brian Roark, Richard Sproat, and Richard William Sproat. 2007. *Computational approaches to morphology and syntax*, volume 4. Oxford University Press.

Carl Rubino. 2005. Reduplication: Form, function and distribution. *Studies on reduplication*, pages 11–29.

Lane Schwartz, Emily Chen, Benjamin Hunt, and Sylvia Schreiner. 2019. Bootstrapping a Neural Morphological Analyzer for St. Lawrence Island Yupik from a Finite-State Transducer. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, pages 87–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver. Association for Computational Linguistics.

Appendix

We provide the MarianNMT configuration settings used for all neural models in this work.

```
--type amun
--dim-vocabs 600 500
--mini-batch-fit -w 3500
--layer-normalization
--dropout-rnn 0.2
--dropout-src 0.1
--dropout-trg 0.1
--early-stopping 5
--valid-freq 10000
--save-freq 10000
--disp-freq 1000
--valid-metrics cross-entropy
--overwrite
--keep-best
--seed 1111
--exponential-smoothing
--normalize=1
--beam-size=12
--quiet-translation
```