Unsupervised morphological segmentation in a language with reduplication

Simon Todd and **Annie Huang**
Department of Linguistics
University of California, Santa Barbara
{sjtodd, anniehuang}@ucsb.edu

Jeremy Needle
jeremyneedle@gmail.com

Jennifer Hay and **Jeanette King**
New Zealand Institute of Language, Brain and Behaviour
University of Canterbury
{jen.hay, j.king}@canterbury.ac.nz

Abstract

We present an extension of the Morfessor Baseline model of unsupervised morphological segmentation (Creutz and Lagus, 2007) that incorporates abstract templates for reduplication, a typologically common but computationally underaddressed process. Through a detailed investigation that applies the model to Māori, the Indigenous language of Aotearoa New Zealand, we show that incorporating templates improves Morfessor's ability to identify instances of reduplication, and does so most when there are multiple minimally-overlapping templates. We present an error analysis that reveals important factors to consider when applying the extended model and suggests useful future directions.

1 Introduction

Unsupervised models that can learn to segment words into morphemes without requiring extensive hand-written rules have two important advantages (see Creutz and Lagus, 2007, for discussion). First, their unsupervised nature allows them to capture a key facet of human morphological learning: learning despite the lack of both direct and negative evidence. Second, their lack of hand-written rules makes them very flexible: they can be deployed in a range of applications, across diverse languages.

However, in order to learn effectively, unsupervised models must make general assumptions about underlying morphological processes, and their success in part reflects the appropriateness of these assumptions for the language(s) under investigation. This can cause the underlying assumptions to become tuned to the morphological processes of high-resource languages used in development and evaluation (Bender, 2009), leading models to overlook processes that do not occur in such languages, even if they are typologically common.

Recent work has highlighted the advantages to such models of incorporating expert linguistic knowledge, such as language-specific morphemes and/or abstract morphological templates (Butler, 2016; Eskander et al., 2016; Godard et al., 2018; Xu et al., 2020). We explore the value added to a standard baseline model, Morfessor (Creutz and Lagus, 2007; Virpioja et al., 2013), by incorporating templates for reduplication, a typologically common but computationally underaddressed process. We conduct a detailed analysis of the successes and challenges in using an enriched model to capture reduplication in Māori (Polynesian), the Indigenous language of Aotearoa New Zealand, which reveals a promising path for unsupervised morphological segmentation of languages with reduplication more broadly.

2 Background

2.1 Unsupervised morphological segmentation

Morphological segmentation aims to identify boundaries within words by splitting them into parts, as in *de* + *forest* + *ation*. In unsupervised approaches, the inventory of parts is inferred from the training data, by identifying the *morphs* – sequences of characters, phonemes, or larger 'atoms' – that recur across words with statistical regularity. There are several models for unsupervised morphological segmentation, many permitting fine-grained structural assumptions about underlying morphological processes (e.g. Goldsmith, 2001; Johnson and Griffiths, 2007; Eskander et al., 2016; Godard et al., 2018; Xu et al., 2018, 2020).

We focus on the Morfessor family of models (Creutz and Lagus, 2007), often used as a baseline due to its extremely simple assumptions. Morfessor uses a Minimum Description Length framework

(Rissanen, 1978): it aims to identify the smallest and simplest set of morphs (the *lexicon*) that generates the training data with highest probability. The lexicon is treated as a bag of morphs, where the cost of adding a given morph to the lexicon in training is based on the complexity of its form as well as the frequency with which it recurs across words. The training data are assumed to be generated from the lexicon by concatenating morphs that are drawn independently from it, with no consideration of contraints based on position, sequencing, or morphosyntactic category. Morfessor is particularly suited to languages that make heavy use of concatenative morphological processes with limited or no phonological alternations. We explore whether it can be expanded to account for reduplication, by extending the Python implementation of Morfessor 2.0 (Virpioja et al., 2013).

2.2 Reduplication and Morfessor

Reduplication is defined by Rubino (2005) as "the systematic repetition of phonological material within a word for semantic or grammatical purposes". Informally, it is often described as a process by which a *reduplicant* phonologically 'copies' part of a *base* to which it is morphologically attached. The reduplicant may copy the entire base, as in the Māori *pakipaki* 'to clap' (from *paki* 'to slap'), or only part of it, as in Māori *nunui* 'big.PL' (from *nui* 'big.SG'). In formal linguistic theory, the reduplicant is commonly treated as a morpheme, RED, which has little or no inherent phonological content, and copies content from the base in order to satisfy prosodic wellformedness templates (e.g. Marantz, 1982; McCarthy and Prince, 1996). In this view, the reduplicant attaches to the base in the same way as any other morpheme would. However, for clarity, we notate these kinds of morphological attachment differently, using $\oplus$ to represent a boundary between a reduplicant and its base, and + to represent all other boundaries.

Rubino (2013) reports that 85% of languages documented in the World Atlas of Language Structures include some productive form of reduplication. Yet, despite its prominence, reduplication is not typically given special treatment in unsupervised approaches to morphological segmentation. For Morfessor, we are only aware of one system incorporating reduplication (Butler, 2016); however, it identifies and rewrites potential instances of reduplication *outside* of Morfessor, following a heuristic, rather than *within* Morfessor, according to statistical evaluation. It treats reduplication as a feature of the data rather than of the probabilistic grammar of the language, limiting the ability to leverage knowledge of reduplication to navigate ambiguity or generalize beyond the training set.[1]

The lack of integrated special treatment of reduplication limits Morfessor's ability to consistently identify reduplicants, due to their variable form. In turn, the repeated failure to isolate reduplicants from their bases limits Morfessor's ability to identify these bases as independent morphs elsewhere, outside of reduplication. The incorporation of special treatment of reduplication into Morfessor thus stands to vastly improve its reliability, not only in reduplicated words but also in general.

2.3 The Māori language

Māori is an ideal test case for four reasons. First, its orthography maps to phonemes unambiguously[2], enabling morphological segmentation to be applied straightforwardly to written words. Second, it has clear atoms for morphological segmentation, as morpheme boundaries typically coincide with the boundaries of (C)V units (Bauer, 1993). Third, its morphology predominantly includes concatenative processes (Krupa, 1968) and makes heavy use of compounding, alongside a few highly productive affixes (Bauer, 1993; Harlow, 1993). Fourth, approximately 25% of its word types include reduplication (often alongside other morphological processes; Todd et al., 2019), implying that it stands to gain a lot from the incorporation of reduplication into morphological segmentation systems.

Māori has many kinds of reduplication (see Keegan, 1996), all requiring the base to contain at least 2 morae, where a syllable with a short vowel has 1 mora and a syllable with a long vowel has 2 (Harlow, 1991). We focus on the 5 most common kinds: *full*, in which the reduplicant copies the whole base; *left-1*, in which it copies the first mora from the base; *left-1L*, in which it copies the first mora and lengthens its vowel; *left-2*, in which it copies the first 2 morae from a base containing at least 3 morae; and *right*, in which it copies the last 2 morae from a base containing 4 morae, where the first syllable has a long vowel (see Table 1).

[1] A direct comparison between our extension to Morfessor and alternative models is left for future work.

[2] Each phoneme is represented by a single character, except for the digraphs ⟨wh⟩ (/f/) and ⟨ng⟩ (/ŋ/). Macrons ⟨ā, ē, ī, ō, ū⟩ designate long vowels.

Kind	Examples
full	*pakipaki*, *whiuwhiu*, *tōtō*
left-1	*nunui*, *hahana*, *huhū*
left-1L	*kākahu*, *mīmiro*, *rērere*
left-2	*huahuaki*, *kuikuia*, *māmāika*
right	*tākaikai*, *hāmamamama*, *ūkuikui*

Table 1: Common kinds of Māori reduplication.

3 Extending Morfessor to reduplication

3.1 General approach

Consistent with the common approach within linguistic theory, we treat all reduplicants as corresponding to one morph, RED, which has no phonological content. We add RED to the lexicon underlying the Morfessor training and testing algorithms, such that identifying a new instance of reduplication allows the algorithms to ignore the form-based component of the cost of the reduplicant, and to reduce its usage-based cost by pooling counts across all other reduplicants in already-identified instances of reduplication. Importantly, we do not assume that all potential instances of reduplication are actual instances of reduplication, either in training (Section 3.2) or in testing (Section 3.3).[3]

We use manually-defined templates to identify potential instances of reduplication, which are assessed by Morfessor for their statistical support as actual instances. The templates are loosely specified, to permit them to capture arbitrary copying in any language. Given the side of reduplicant attachment and the minimum size of the base, potential instances of reduplication are flagged by string comparison of adjacent sequences of atoms (phonemes, syllables, etc.). Additional specifications can be added on a language-by-language basis, leveraging expert knowledge for tighter control; these may include constraints on size or shape of the reduplicant or base, or even systematic alternations between correspondents in the reduplicant and base (e.g. Māori *left-1L* reduplication, *kākahu*).

For Māori, we define three mutually-exclusive templates as generalizations over the kinds and constraints described in Section 2.3. In all templates, the base must be at least bimoraic. In the full-reduplication template, the reduplicant and base must be the same size; in the left-reduplication template, the reduplicant may be any size smaller than the base, and, if monosyllabic, may consist of a single syllable that lengthens the vowel of its correspondent in the base; and in the right-reduplication template, the reduplicant must be at least bimoraic and shorter than the base, which must have a long vowel in the first syllable. Because the templates are mutually exclusive, each may be included in the model or excluded, independent of the others.[4]

We also make the (Māori-specific) assumption that the base must be morphologically simplex (following Krupa, 1968). Thus, when Morfessor commits to analyzing a word as an instance of reduplication (e.g. of analyzing *tākaikai* as tākai $\oplus$ RED), we block it from considering any future placement of boundaries within the minimal base (tākai).

3.2 Training models with reduplication

Training in Morfessor uses the recursive splitting algorithm (henceforth, RS; Creutz and Lagus, 2002). For a given input, RS evaluates all possible analyses that split the input into two parts, as well as the analysis that leaves it unsplit. It chooses the analysis for which the associated parameter update permits lowest-cost generation of the training data. If the chosen analysis splits the input into parts, the algorithm recurses to evaluate analyses of each part; otherwise, it moves on to evaluate the next input. It cycles through all words in a training set once per epoch, and repeats until the epoch-wise decrease in cost falls below a threshold.

We extend RS to consider reduplication. When the analysis under consideration splits a potential reduplicant at the edge of the input from its apparent base (e.g. nu $\oplus$ nui), we consider an analysis that replaces the reduplicant with RED (RED $\oplus$ nui). This analysis will be chosen if it is associated with lower cost than any alternative.

We do not automatically consider an analysis involving reduplication if the potential reduplicant is not at the edge of the input, as in many words involving compounding or affixation (e.g. *whārarahi*, whā + [RED $\oplus$ rahi]). If the compound component or affix (*whā*) is split off first, leaving the reduplicant at the edge of one part (*rarahi*), we consider reduplication as above. Otherwise, we only con-

[3] Code for our approach, consisting of a patch to Morfessor 2.0 (Virpioja et al., 2013), is available at https://github.com/sjtodd/morfessoRED. At the time of writing, detailed documentation is still under development.

[4] The full-reduplication template assumes that the 'default' side on which the reduplicant attaches is the left, unless the left-reduplication template is not included in the model and the right-reduplication template is, in which case it assumes attachment on the right for parsimony.

sider reduplication if RS finds no binary-splitting analysis that is better than leaving the input unsplit (whārarahi), in which case we evaluate whether the ternary-splitting analysis implied by reduplication is associated with lower cost than the unsplit analysis. This allows reduplication to be leveraged as a cue to the presence of compounding or affixation, but ensures that we do not overgeneralize by relying on this cue too strongly.

Finally, if it is ambiguous whether an edge-aligned reduplicant corresponds to full-reduplication or another kind of reduplication (e.g. whether *huahuaki* is [RED ⊕ hua] + ki or RED ⊕ huaki), we leave both options open by neither enforcing nor restricting a boundary placement after the apparent full-reduplication base (hua). If RS goes on to place a boundary here, we analyze it as full-reduplication; otherwise, we analyze it as the other kind of reduplication. This is consistent with the use of loosely-specified templates that allow arbitrary copying.

3.3 Applying models to seen and unseen data

In testing, Morfessor uses the segmentation obtained from RS if the word was observed in training. Otherwise, it uses the Viterbi algorithm (Viterbi, 1967) to find the optimal path through potential boundary sites (see Virpioja et al., 2013).

The standard Viterbi algorithm proceeds 'horizontally' through potential boundary sites in a word, identifying at each site the optimal previous site to have come from in left-to-right order (Figure 1(a)). We extend the algorithm by adding a 'vertical' dimension, which holds partial analyses matching different reduplication templates (Figure 1(b)). At each potential boundary site in the word, the set of 'horizontal' candidates for optimal previous site is augmented with a small number of directly neighboring 'vertical' sites representing partial analyses based on reduplication templates. As in RS, the reduplicant is replaced by RED in the evaluation of reduplication partial analyses.

4 Experiments

4.1 Data

The models were trained on a set of 19,595 word types from the Te Aka Dictionary (Moorfield, 2011), with all kinds of morphological structures (i.e. not just reduplication). To form this set, we took all headwords, together with their listed inflections. When a headword was composed of words separated by whitespace or hyphens, we split it into components. We then removed (capitalized) proper nouns, because they are likely to be place name borrowings, or are otherwise unlikely to follow the same morphological grammar as other words.

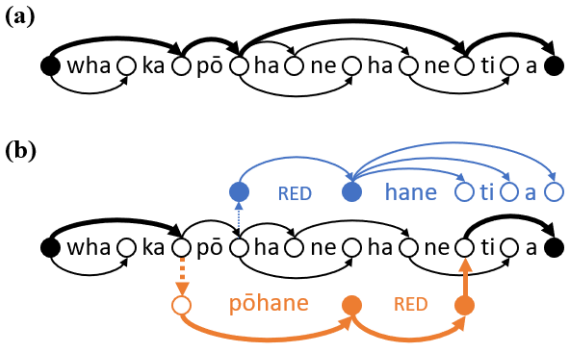

Figure 1: Segmentation traces for *whakapōhanehanetia* from the Viterbi algorithm. Solid circles indicate required boundaries. We extend the standard algorithm (a) by adding a dimension for reduplication paths (b).

Data	*full*	*left-1*	*left-1L*	*left-2*	*right*
Training	816	588	169	786	1191
Test	747	314	79	56	693

Table 2: Distribution of words across different kinds of reduplication. The training data also contains 16,045 other words, many of which combine reduplication with compounding and/or affixation.

The models were tested on a set of 1,889 word types categorized by Keegan (1996, Appendices A–D) as clear instances of the kinds of reduplication under investigation. Based on Keegan's categorization, we inferred a gold standard segmentation for each word. We removed words where the apparent base was likely morphologically complex, as determined by consisting of more than 4 morae or more than 3 syllables (cf. Krupa, 1968; de Lacy, 2003), to allow us to focus on the ability to capture reduplication without influence of other morphological processes. We cross-referenced the final test items with the Te Aka Dictionary (Moorfield, 2011) in order to ensure consistency in the identification of long vowels. 83.5% of the test items were in the dictionary (i.e. the training data).

Table 2 shows the distribution of words across reduplication templates in the two datasets.

4.2 Metrics

We report four metrics: accuracy, recall, and two versions of precision. Each metric is macro-

Segmentation	*Acc.*	*Rec.*	*Prec.0*	*Prec.1*
tākai ⊕ kai	1	1	1	1
tā + kaikai	0	0	0	0
tā + kai ⊕ kai	0	1	0.5	0.5
tākaikai	0	0	0	1

Table 3: Example metrics for various segmentations of *tākaikai*, where + designates a boundary and ⊕ designates the gold boundary between reduplicant and base. This designation is for ease of reference only; all predicted boundaries are treated alike in calculations.

Model	Acc.	Rec.	Prec.0	Prec.1
original	0.23	0.34	0.28	0.59
extended	**0.83**	**0.98**	**0.91**	**0.91**

Table 4: Test metrics for original Morfessor (no reduplication templates) and extended model (all templates).

averaged, i.e. calculated on a per-word basis and then averaged across all words in the test set. All metrics are calculated based on the morph boundaries contained within the segmentation of a word. Since the words in the test set have morphologically simplex bases for reduplication, the gold standard segmentation contains only a single boundary.

For a given word, accuracy (*Acc.*) is 1 if the model predicts a single boundary matching the gold boundary, and 0 otherwise. Recall (*Rec.*) is 1 if the model's predicted boundaries include the gold boundary, and 0 otherwise. When the model predicts $n \geq 1$ boundaries, both versions of precision (*Prec.0* and *Prec.1*) are $1/n$ if one of those boundaries is the gold boundary, and 0 otherwise. When the model leaves a word unsplit, predicting no boundaries for it, *Prec.0* is 0, while *Prec.1* is 1.[5] The metrics are illustrated in Table 3.

4.3 Overall effects of reduplication templates

Our results show that incorporating reduplication templates leads to substantial improvements over the original Morfessor model (see Table 4). The original model has two main issues. First, it predicts no boundaries for a lot of test items (571 items / 30.2%). Second, the boundaries it does predict usually do not match the gold boundary; for example, it predicts a single boundary for 1,094 items (57.9%), but this only matches the gold boundary 39.9% of the time (437 items). Even when it (incorrectly) predicts multiple boundaries (224 items), the gold boundary is not among them 9.4% of the time (21 items). By contrast, the extended model predicts no boundaries for very few test cases (14 items / 0.7%) and a single boundary for most (1,579 items / 83.6%), matching the gold boundary 99.4% of the time (1,570 items). It (incorrectly) predicts multiple boundaries slightly more often than the original (296 items), but it is rarer for the gold boundary not to be among them (13 items / 4.4%).

[5] *Prec.1* is the version of precision in the Morfessor 2.0 Python implementation (Virpioja et al., 2013). It artificially rewards models that leave words unsplit; introducing *Prec.0* allows us to make comparisons that account for this. To avoid ambiguity of interpretation resulting from the presence of two versions of precision, we do not calculate an *F*-score.

n	Templates			*Acc.*	*Rec.*	*Prec.0*	*Prec.1*
0	-F	-L	-R	0.23	0.34	0.28	0.59
1	-F	+L	-R	0.34	0.44	0.39	0.67
1	-F	-L	+R	0.47	0.53	0.50	0.77
1	+F	-L	-R	0.48	0.83	0.65	0.72
2	+F	-L	+R	0.54	0.87	0.70	0.75
2	+F	+L	-R	0.59	0.97	0.78	0.79
2	-F	+L	+R	0.65	0.71	0.68	0.86
3	+F	+L	+R	**0.83**	**0.98**	**0.91**	**0.91**

Table 5: Test metrics for models with different numbers (n) and types of reduplication templates.

4.4 Effects of individual templates

Table 5 gives a comparison of models with different combinations of templates. It clearly shows that all templates are needed in order to attain best model performance. It also shows that performance generally increases with the number of templates included, especially if they cover a diverse and minimally-overlapping range of situations.

When the model contains only a single reduplication template, its performance is largely driven by the prevalence of that template in the test data. When the model contains two templates, performance is no longer driven entirely by prevalence, because the templates may interact: both may match the same test item and compete over it, while neither matches a large class of other items. For example, the two-template model containing right- and full-reduplication templates performs worse than the model containing left- and full-reduplication templates, despite there being more *right* test items than *left* test items.

The templates interact here for two main reasons. The full-reduplication template interacts with any other because it allows the reduplicant to attach on

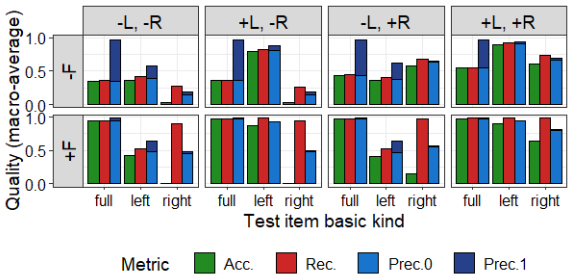

Figure 2: Performance metrics for models with different reduplication templates on test items with different basic kinds of reduplication.

a 'default' side set by the other template. Combining the left- and full-reduplication templates causes competition over *left-2* test items (e.g. RED ⊕ huaki vs. [RED ⊕ hua] + ki for *huahuaki*) and coercion of all *right* test items to the full-reduplication template (e.g. tā + [RED ⊕ kai] for *tākaikai*), and vice versa for combining the right- and full-reduplication templates. The full- and right-reduplication templates also interact because they both allow reduplicants of the same size. Combining them in a single model causes some *right* test items to be coerced to the full-reduplication template (e.g. tā + [RED ⊕ kai] for *tākaikai*), while *left-1* items (e.g. *nunui*) are left unmatched to any template.

Above and beyond such interactions, a consistent property of the full-reduplication template shines through: it consistently closes the gap between the two versions of precision. This suggests that, in the absence of relevant templates, *full* test items such as *pakipaki* are typically predicted not to contain a boundary. To a human, this failure to predict a boundary is remarkable, as full reduplication is a highly salient cue to morphological structure.

4.5 Kinds of reduplication captured

To confirm the idea that the model performs well with the addition of new templates because they allow more (and more diverse) test items to be matched to a template, we explored performance across items representing different kinds of reduplication. The results (Figure 2) confirm three key patterns noted earlier. First, the model containing all templates performs best because it can capture all kinds of reduplication well. Second, models generally perform better on a given kind of reduplication when they include the corresponding template; for example, *left* items are best captured if models contain the left-reduplication template. Third, interactions between templates can cause competition, reducing performance on certain kinds of items. For example, when the model contains the right-reduplication template but not the left-reduplication template, accuracy and precision for *right* items decrease with the inclusion of the full-reduplication template, as discussed in Section 4.4.

There is also a fourth pattern, which elaborates on the observation that performance generally increases with the number of templates included. In Figure 2, it is clear that the increase is not driven just by the diversification of templates, but also by the increased statistical support that more templates bring for the recognition of RED as a morph. Since the same RED morph is shared across all templates, increased ability to identify RED in test items matching one template may also increase the ability to identify it in test items matching a different template. This can be seen in the way that adding the right-reduplication template to a model already including the left-reduplication template causes an improvement on *left* test items.

The same patterns are revealed by detailed breakdowns within a given kind of reduplication, as shown for left-reduplication in Figure 3. This breakdown also shows that different subkinds exhibit the patterns to different extents. For example, *left-1* items benefit more from the inclusion of the left-reduplication template than *left-1L* items do, because the CV̄ reduplicant in *left-1L* cases typically has the same form as one of several (fossilized) prefixes that recur across a number of words (Krupa, 1968; Harlow, 2007), so it has sufficient statistical support to be segmented away from the base without recourse to reduplication. Similarly, *left-2* items are uniquely affected by an interaction that sees them coerced to a full-reduplication template (e.g. [RED ⊕ horo] + i instead of RED ⊕ horoi for *horohoroi*), because only they have a bimoraic reduplicant that is identical to its correspondent in the base.

These results show that careful thought is needed when adding reduplication templates to the model. If templates are attuned to distinct reduplication patterns in the language, they can allow the model to perform well both specifically, on items matching these templates, and generally, across all items containing reduplication. But, if the templates are too general or too numerous, they can interact with each other and endanger the ability to capture particular subsets of test items.

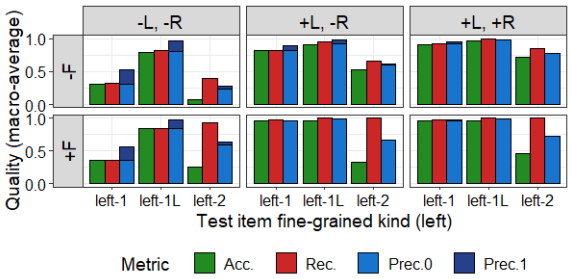

Figure 3: Performance metrics for models with different reduplication templates on test items with different kinds of left-reduplication.

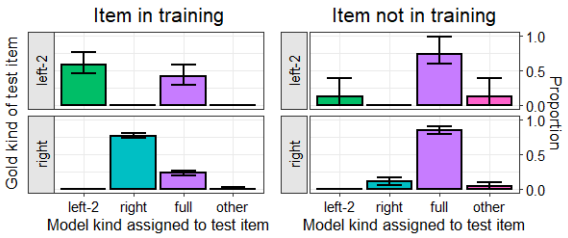

Figure 4: Partial confusion matrix for classification of *left-2* and *right* test items, based on whether the item was in the training data.

5 Error analysis and improvements

5.1 Coercion to full reduplication

As previously noted, different reduplication templates can interact (Section 4.4), affecting model performance on items with certain kinds of reduplication (Section 4.5). In particular, including the full-reduplication template can limit accuracy and precision on *left-2* and *right* test items, as these items are coerced to match the full template rather than their own. Since the best model includes all templates, it shows these interactions: coercion of *left-2* and *right* test items to the full-reduplication template accounts for 88.4% of errors (221 of 250). Nevertheless, because there are so many *full* items in the test set, and because the identification of RED in these highly salient items offers increased statistical support for the identification of RED elsewhere, it is still better to include the full-reduplication template than not, as shown in Table 5.

One strategy for reducing coercion of *left-2* items to the full-reduplication template might be to require that, when the left-reduplication template is matched, the base must be longer than the reduplicant. Currently, the base must contain at least 2 morae, but it is not required to be longer if the reduplicant is bimoraic. However, this would likely cause problems for items involving full reduplication alongside compounding or affixation, such as *tomotomokanga* ([RED ⊕ tomo] + kanga), which are omitted from the current test set but frequent in the language. These would only be able to be recognized as containing full reduplication if the part of the word that is not reduplicated is split off prior to the reduplication template being matched, which is unlikely as RED has more statistical support than any single affix or compound component.

This strategy would not apply to *right* items, as that template already requires that the base be longer than the reduplicant. These items are coerced to the full-reduplication template mainly because the initial CV̄ has the same form as one of several prefixes (cf. Section 4.5). A strategy for mitigating this might be to introduce a penalty for overzealous splitting off of monosyllabic morphs. The size of such a penalty would have to be tuned carefully so that an initial CV̄ syllable can still be split off outside of *right* items, where it has no better alternative analysis than as a prefix.

5.2 Coercion-blocking and Viterbi decoding

As described in Section 3.3, segmentations are obtained for test items in different ways. For items observed in training, the segmentation obtained from RS is used, while for items not observed in training, a segmentation is obtained from the Viterbi algorithm. As shown in Figure 4, it is test items that were not observed in training that show the most coercion to the full-reduplication template.[6]

RS blocks coercion to the full-reduplication template because it commits to boundaries one at a time. In RS, a *right* item such as *tākaikai* will usually have its first boundary placed in-between the reduplicant and base (tākai ⊕ RED), which commits the algorithm to a right-reduplication template and prevents any further boundaries from being placed within the base (tākai). The only way RS could end up coercing the item to the full-reduplication template (tā + [RED ⊕ kai]) is if it placed the first boundary after the initial CV̄ syllable (tā + kaikai) instead, but this is unlikely because the CV̄ syllable is much less common than RED and thus has less statistical support for being split off.

By contrast, the Viterbi algorithm does not

[6]The extended model still outperforms original Morfessor on items not observed in training, in spite of the large amount of coercion, through improved treatment of other kinds of reduplication. Metrics on untrained items (*Acc.* / *Rec.* / *Prec.0* / *Prec.1*) for original: 0.29 / 0.49 / 0.38 / 0.41; for extended: 0.51 / 0.94 / 0.72 / 0.72.

Figure 5: Partial confusion matrix for classification of *right* test items from the training data, based on whether the base was separately observed in training.

block coercion to the full-reduplication template because it does not commit to boundaries independent of each other. When evaluating the best segmentation for a *right* item such as *tākaikai*, it will typically end up comparing the complete full-reduplication segmentation (tā + [RED ⊕ kai]) with the complete right-reduplication segmentation (tākai ⊕ RED). Because the initial CV̄ syllable (tā) recurs across words much more than the actual base (tākai), the full-reduplication template will typically have more statistical support.

This difference suggests two possible strategies for improving model performance. One is to train the model on as many different word types as possible, increasing the chance that any given test item will have been observed in training and will therefore get its segmentation through RS. An alternative strategy is to develop a recursive segmentation algorithm that can be used in testing without triggering changes to trained model parameters.

5.3 Independence of the base

While *right* test items that were observed in training are coerced to the full-reduplication template much less often than those that were not observed, they are still coerced sometimes (see Figure 4). As shown in Figure 5, RS coerces *right* test items to the full-reduplication template more often when the base for reduplication was not separately observed in training. This is because it considers the statistical support for both word-parts created by the insertion of a boundary: the base and the reduplicant. When the base is listed independently in the training set, both parts have some support, and the segmentation is likely to be accepted. But when the base is not listed in the training set – for example, for the word *pānekeneke* – only RED has support, and the algorithm penalizes the right-reduplication segmentation for having to add the base to the lexicon. By contrast, the placement of a boundary after the initial CV̄ syllable (pā + nekeneke) can yield two word-parts that are already listed in the training set (*pā* and *nekeneke*), offering a penalty-free alternative segmentation. Because RS inherits its pre-identified substructure of word-parts, and because one of the parts in this case is likely to have been pre-identified as an instance of full reduplication (RED ⊕ neke), the alternative segmentation amounts to coercion to the full-reduplication template. Both the right-reduplication segmentation and the alternative segmentation therefore gain equally strong statistical support from RED, and the alternative segmentation typically wins because it does not enforce a new-morph penalty.

One strategy that might limit errors when the base for reduplication is not in the training set is to alter RS to block the inheritance of pre-identified substructure pertaining to a reduplication template. However, it is possible that this would limit the ability to use reduplication as a cue to the internal structure of a compound such as *pōpōroroa*.

6 Experiments on complex words

To see how incorporating reduplication templates affects segmentation of morphologically complex words, we now compare the extended model (all templates) with the original (no templates) on a broader subset of training data, examining their agreement with fluent-speaker segmentations.

Data. We analyze model segmentations of 4,213 words of 3+ morae on which two fluent speakers of Māori agreed. None of these words contain long vowels, since we have documented elsewhere that these speakers show an extreme sensitivity to long vowels (Todd et al., 2019; Panther et al., under review); for example, they segmented *hāro* (which is morphologically simplex) as hā + ro, and routinely split off the initial long-vowel syllable of *right* reduplication items, as in kā + witi ⊕ witi and hā + upaupa. As such, the dataset contains no instances of *right* or *left-1L* reduplication, which require a long vowel, and reduced instances of *left-2* reduplication, for which the reduplicant may contain a long vowel. It also contains no instances of *full* reduplication alone (e.g. *pakipaki*), as the original data collection purposes did not require segmentations for words with transparent structures.

Methods. We treat the fluent-speaker segmentations as a reference set, such that performance metrics describe *agreement* between models and speakers. This approach is imperfect; for example, the speakers failed to segment a number of instances of *left-1* reduplication (e.g. ririki instead of ri ⊕ riki)

and missegmented others (e.g. hoho + rea instead of ho ⊕ hore + a). In particular, due to the concentration of speaker errors on reduplicated words and the omission of a large host of reduplicated words from the dataset, this approach under-rewards models that correctly handle reduplication. Nevertheless, it gives a sense of how the models perform in more complex settings than our previous test set.

Results. On this subset, the models have very similar accuracies (agreement with speakers): 0.68 for the original model, and 0.70 for the extended model. Thus, incorporating reduplication templates does not hurt model performance in general.

Figure 6 breaks down the results by morphological process for the 3,380 words judged by the speakers to involve affixation and/or compounding. The extended model performs much better than the original on complex reduplicated words, generalizing advantages seen previously for simple words. While it performs slightly worse than the original on complex non-reduplicated words, particularly affixed words, this decrease is small relative to the increased performance on reduplicated words, and does not decrease performance overall.

Error analysis. There are 228 words for which the extended model is discrepant with the speakers but the original is not. We could unambiguously infer a correct segmentation from Te Aka (Moorfield, 2011) for 191 words, highlighting three main reasons for discrepancies. First, the speakers failed to segment reduplicant in 39 reduplicated words. This reflects imperfections of the reference set, not failures of the model. Second, the model identifies reduplication in 31 non-reduplicated words (e.g. ni ⊕ nia instead of nini + a). False positives like these are to be expected, and can be tolerated because they are few in relation to the true positives. Third, the model undersegments in 102 words, including failing to segment out affixes in 71 words. This is not a major cause for concern, as the undersegmentation is not systematic: the missed affixes are correctly segmented in other words.

7 Conclusion

We have described a method to incorporate abstract reduplication templates into the Morfessor baseline model of unsupervised morphological segmentation (Creutz and Lagus, 2007; Virpioja et al., 2013). Our test on Māori shows three main results. First, incorporating templates allows Morfessor to better identify instances of reduplication. Second, the more distinct templates incorporated, the better the model performs. Third, the benefits of incorporating additional templates are strongest for items matching those templates, but also present for items matching other templates, due to the pooling of statistical support for the reduplicant morph, RED.

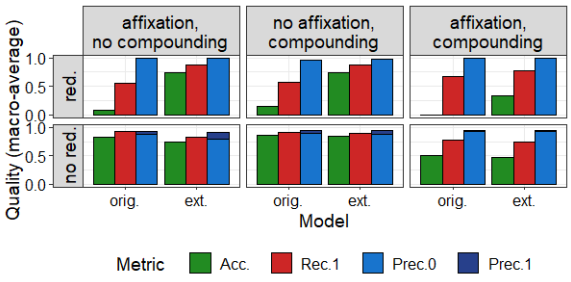

Figure 6: Performance metrics for original and extended Morfessor models against fluent-speaker segmentations of 3,380 words involving affixation and/or compounding, with (top) or without (bottom) reduplication.

We have also discussed factors that should be considered when applying the extended model. First, care should be taken to minimize interactions between templates, to avoid competition that coerces multiple kinds of reduplication to the same template. Second, the training set should be as large and as similar to the test set as possible, because coercion between templates is more prevalent in the Viterbi algorithm used for untrained items than it is in the recursive algorithm used for trained items. Third, the training set should include both reduplicated forms and their (apparent) bases of reduplication, as excluding the base can preclude it from being identified in the reduplicated form, which can in turn increase the risk of coercion to an incorrect reduplication template.

Our results clearly show the value of incorporating expert linguistic knowledge into unsupervised morphological segmentation. We have shown how this improves segmentation of reduplicated words in Māori, while still permitting accuracy on non-reduplicated words. While we have focused on Māori, we expect performance gains to transfer to other Polynesian languages with similar reduplication templates, and we expect the higher level modeling approach and insights to extend more broadly to any language that has productive reduplication processes. Given the high typological prominence of reduplication (Rubino, 2013), the incorporation of reduplication templates offers a promising avenue for improving the cross-linguistic adequacy of unsupervised morphological segmentation.

Acknowledgments

We thank the three anonymous reviewers for their feedback. We are grateful to Te Puawai Wilson-Leahy and Tamahou Thoms for providing fluent-speaker segmentations, and to John C. Moorfield for permission to use data from Te Aka. This work was supported by funding from Te Pūtea Rangahau a Marsden / The Marsden Fund (UOC1502).

References

Winifred Bauer. 1993. *Maori*. Routledge, London.

Emily M. Bender. 2009. Linguistically naïve != language independent: Why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 26–32.

Steven R. Butler. 2016. *Infixer: A Method for Segmenting Non-Concatenative Morphology in Tagalog*. Unpublished MA thesis, City University of New York.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):1–34.

Paul de Lacy. 2003. Maximal words and the Maori passive. In *Proceedings of AFLA VIII: The eighth meeting of the Austronesian Formal Linguistics Association*, volume 44, pages 20–39, Cambridge, MA. MIT Linguistics Department.

Ramy Eskander, Owen Rambow, and Tianchun Yang. 2016. Extending the use of adaptor grammars for unsupervised morphological segmentation of unseen languages. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, pages 900–910.

Pierre Godard, Laurent Besacier, François Yvon, Martine Adda-decker, Gilles Adda, Hélène Maynard, Annie Rialland, and Inria Grenoble. 2018. Adaptor grammars for the linguist: Word segmentation experiments for very low-resource languages. In *Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 32–42.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Ray Harlow. 1991. Consonant dissimilation in Maori. In Robert Blust, editor, *Currents in Pacific Linguistics: Papers in Austronesian Languages and Ethnolinguistics in honour of George W. Grace*, pages 117–128. Australian National University, Canberra.

Ray Harlow. 1993. Lexical expansion in Maori. *Journal of the Polynesian Society*, 102(1):99–107.

Ray Harlow. 2007. *Māori: A Linguistic Introduction*. Cambridge University Press, Cambridge.

Mark Johnson and Thomas L. Griffiths. 2007. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems 19*, pages 641–648.

Peter Julian Keegan. 1996. *Reduplication in Maori*. Unpublished MA thesis, University of Waikato.

Victor Krupa. 1968. *The Maori Language*. Nauka, Moscow.

Alec Marantz. 1982. Re reduplication. *Linguistic Inquiry*, 13(3):435–482.

John J. McCarthy and Alan S. Prince. 1996. Prosodic morphology. In John A. Goldsmith, editor, *The Handbook of Phonological Theory*, chapter 9, pages 283–305. Blackwell, Malden, MA.

John C. Moorfield. 2011. *Te Aka: Māori-English, English-Māori Dictionary*, 3rd edition. Pearson, Auckland.

Forrest Panther, Wakayo Mattingley, Jennifer Hay, Simon Todd, Jeanette King, and Peter Keegan. under review. Morphological segmentations of non-Māori speaking New Zealanders match proficient speakers.

Jorma Rissanen. 1978. Modelling by shortest data description. *Automatica*, 14:465–471.

Carl Rubino. 2005. Reduplication: Form, function and distribution. In Bernhard Hurch, editor, *Studies on Reduplication*, pages 11–30. Mouton de Gruyter, Berlin.

Carl Rubino. 2013. Reduplication. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas on Language Structures Online*, chapter 27. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Simon Todd, Jeremy Needle, Jeanette King, and Jennifer Hay. 2019. Quantitative insights into Māori word structure. Paper presented at the Annual Meeting of the Linguistic Society of New Zealand.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline. Technical report, Department of Signal Processing and Acoustics, Aalto University, Helsinki.

Andrew J. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Hongzhi Xu, Jordan Kodner, Mitchell Marcus, and Charles Yang. 2020. Modeling morphological typology for unsupervised learning of language morphology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6672–6681.

Hongzhi Xu, Mitchell Marcus, Charles Yang, and Lyle Ungar. 2018. Unsupervised morphology learning with statistical paradigms. In *Proceedings of COLING 2018, the 27th International Conference on Computational Linguistics*, pages 44–54.