

THE FIRST INTERNATIONAL CONFERENCE

ON

**LANGUAGE TECHNOLOGIES FOR
LOW-RESOURCE LANGUAGES**

LaTeLL'2026

PROCEEDINGS

Edited by:

Salima Lamsiyah, Tharindu Ranasinghe, Saad Ezzini,
Ernesto Luis Estevanell-Valladares, Ruslan Mitkov

Fes, Morocco
September 28–30, 2026
<https://latell.org/2026/>

**THE FIRST INTERNATIONAL CONFERENCE
ON LANGUAGE TECHNOLOGIES FOR
LOW-RESOURCE LANGUAGES
(LaTeLL'2026)**

PROCEEDINGS

28–30 September, 2026
Fes, Morocco
<https://latell.org/2026/>

© 2026 The authors of the individual papers, for their respective contributions.

The individual papers in this volume are licensed by their authors under a Creative Commons
Attribution 4.0 International License (CC BY 4.0).

LaTeLL'2026 is organised by the University of Alicante (Spain), Lancaster University (UK), King Fahd University of Petroleum and Minerals (Saudi Arabia) and the University of Luxembourg, with the local support of Mohammed VI Polytechnic University (Morocco).

The organisers gratefully acknowledge the support of AutoTrust AI, Diamond Sponsor of LaTeLL'2026.

Automating Mwotlap morphology using formal grammar

Fabio Meroni

University of the Basque Country
fmeroni001@ikasle.ehu.eus

Alexandre François

Lattice, CNRS-ENS-PSL-USN
alexandre.francois@ens-psl.eu

Max Silberztein

Marie and Louis Pasteur University
max.silberztein@univ-frcomte.fr

Abstract

This paper presents a rule-based computational formalization of vowel alternations in Mwotlap morphology using NooJ. Mwotlap displays several morphophonological processes at the prefix-root boundary, including vowel copy, epenthesis, transfer, elision, and blockage. To model these first three, which cannot be derived by simple string concatenation within NooJ's finite-state framework, we introduce a new string-edit operator which swaps adjacent segments. This operator unifies vowel copy, insertion, and transfer under a single formal mechanism while preserving linguistic generalization.

1 Introduction to non-concatenative morphology

Morphology is not always strictly linear (as with suffixes and prefixes), and its interaction with phonology can give rise to complex surface patterns, a phenomenon known as non-linear or non-concatenative morphology (Kroeger, 2005, pp. 304ff). In many languages, phonotactic constraints and prosodic requirements intersect with morphological boundaries in ways that trigger processes such as vowel epenthesis and vowel copying. These strategies may serve to repair illicit consonant clusters, satisfy minimal word conditions, or preserve preferred syllable structures, often yielding outputs that reflect both the local phonological environment at morpheme boundaries and the morphological category of the affixes involved.

Olson (2001) described that in Mono (Ubangian, Democratic Republic of the Congo), a well-

formed word needs to contain at least two morae. To appear in isolation, a monomoraic root thus requires the addition of an extra syllable, which takes the form of a copy of the root vowel (1a). Likewise, obstruent-liquid consonant clusters are broken by inserting an epenthetic vowel, whose quality and tone are borrowed from the following vowel (1b). And if such a cluster is found in a monomoraic root in isolation, like *gré* in (1c), then the two rules operate in sequence, resulting in the insertion of two vowels (Hall, 2006, p. 6). Similarly, Kekchi (Mayan, Guatemala) avoids CC clusters at morpheme boundaries with the insertion of a copied vowel (Campbell, 1974, p. 271).

- (1) a. $ʒī \rightarrow īʒī$ 'tooth'
b. $gàfrū \rightarrow gǎfūrū$ 'mortar'
c. $gré \rightarrow égééré$ 'big'

Another phonotactic constraint, this time of Lucazi (Bantu, Angola and Zambia), requires every word to end with a vowel (Fleisch, 2000). If such a vowel is not brought by the root itself, it must be added via epenthesis. Its default form is /a/ (2a) but, when the preceding morpheme is a lexical root, the suffix borrows its quality from the previous vowel (Marten & Chongo Kula, 2002, p. 4) (2b).

- (2) a. *tu-na-fúm-in-a*
SUBJ.1PL-ANT-come-APPL-FV
'we came for (it)'
b. *tu-na-fúm-u*
SUBJ.1PL-ANT-come-FV
'we came'

A single language may employ multiple repair strategies depending on the phonological environment. This study examines such vowel rules in Mwotlap (Oceanic, Vanuatu), a language spoken by 2100 people on Mota Lava, one of the Banks Islands in northern Vanuatu. Its vowel rules are described in François (1999, 2000, 2001), and will be summarized here.

Mwotlap has a strict (C)V(C) syllable structure, which disallows tautosyllabic consonant clusters. If a lexical root has an underlying shape CCVC, phonotactic rules entail that its first two Cs must be distributed across two distinct syllables. In case the root combines with a CV- prefix, the resulting pattern #CV-CVCVC allows the CCVC root to surface unchanged, as in (3a). But in the absence of a supporting prefix, the #CC onset must be broken by an epenthetic vowel, whose quality is borrowed from the following syllable (3b).

- (3) a. *na-syip* < *nA- + syip*
 ART-waterfall
 ‘a/the waterfall’
- b. *siyip* < *syip*
 waterfall
 ‘waterfall’

Another rule, *vowel copy*, concerns several “copiable” prefixes, including the article *na-* that we saw in (3a). When the following root starts with a single C, the prefix alters its underlying vowel to borrow that of the following syllable. Thus in (4), the combination of *na-* with the noun *qoyqoy* ‘dream’ surfaces as a form *no-qoyqoy*, showing vowel copy:

- (4) *no-qoyqoy* < *nA- + qoyqoy*
 ART-dream
 ‘a/the dream’

Finally, a small number of irregular C-initial roots do not trigger vowel copy, but *vowel transfer*. That is, instead of just lending its quality to the vowel of the prefix, the first vowel of the root migrates to the previous syllable, replacing the (copiable) prefix’s underlying vowel – as in (5). Vowel transfer often involves high vowels /i/ or /u/, yet it is not entirely predictable; it is a property of a handful of lexemes.

- (5) *nu-wmet* < *nA- + wumet*
 ART-bowl
 ‘a/the bowl’

When vowel copy, epenthesis, and transfer co-occur within a single morphological system, any computational model must be able to distinguish the three processes, apply them conditional on the radical’s phonological shape, and remain general enough to avoid enumerating each vowel quality as a separate rule. The challenge is therefore to find a level of formalization that is both expressive enough to capture the phonological generalization and constrained enough to remain computationally tractable, that is, to stay within a class of formal grammars that guarantees decidability and linear-time processing.

This paper presents a possible solution to this problem, achieved with a single bounded stack operator conceived in the domain of finite-state Natural Language Processing (NLP). The project presented here aimed specifically at formalizing Mwotlap morphology for subsequent text and corpus analysis, and succeeded with the use of such a tool. In the following pages we present the software environment chosen to develop this idea, within a theoretical framework and a methodological choice that favored linguistic soundness above the aesthetics of the results.

2 Theoretical and methodological background

Today, most NLP projects rely on empirical methods (statistical, deep learning, LLMs) and thus do not require linguists to develop hand-crafted linguistic resources to perform the various tasks they deploy. However, studying and building formalized descriptions of natural languages remains a legitimate scientific goal in and of itself. Ultimately, accurate descriptions of natural languages enable NLP applications to perform more reliably and in a more controlled manner. Most importantly, NLP applications should attempt to emulate the theory that underlies the language’s structures, as defined by the linguist independently of any computational implementation (Silberstein, 2024). Rule-based approaches remain a principled choice for this goal, precisely because they make the linguist’s theoretical commitments explicit and verifiable. The significant results this approach has produced in recent work (e.g. Barclay, 2026)

Prefix	Property	Prefixed to
<i>nA-</i>	Article	
<i>bE-</i>	Adverbializer	Nouns
<i>lE-</i>	Locativizer	
<i>mE-</i>	Perfect	
<i>nE-</i>	Stative	Verbs, nouns, adjectives
<i>tE-</i>	Future	
<i>tE-</i>	Adjectivizer	Locatives
<i>vE-</i>	Enumerator	Numerals

Table 1: The copiable prefixes of Mwotlap. Capital letters mark the vowel as variable.

is itself evidence that linguistic explicitness and computational rigor are not in conflict.

The NooJ linguistic software (Silberstein, 2016) allows linguists to formalize a variety of phenomena (orthography, morphology, syntax and semantics), using the four types of Chomsky-Schützenberger’s (1963) hierarchy of grammars (regular, context-free, context-sensitive and unrestricted grammars). In the present study, we are focusing on a particular aspect of Mwotlap morphology concerning vowel rules, following the linguistic description by François (1999, 2000, 2001).

Non-concatenative morphological processes have been handled in other finite-state frameworks. In XFST (Beesley & Karttunen, 2003) and Foma (Hulden, 2009), such operations can be modelled via compile-replace or flag diacritics (see Amsalu & Demeke, 2006; Cohen-Sygal & Wintner, 2006; Wintner, 2014 inter alia). Before that, two-level morphology, originally proposed by Koskeniemi (1983) and later extended, addressed similar phenomena through the interaction of surface and lexical levels constrained by correspondence rules (see Kay, 1987; Trost, 1990; Pisceldo et al., 2008 inter alia).

Both traditions have produced substantial descriptive coverage across typologically diverse languages, but in a different methodological orientation. An XFST or two-level implementation would require the linguist to write a separate transducer for each non-concatenative process, which is then recomposed with the lexicon externally. Our NooJ-based solution, instead, expresses the operation inline, within the paradigm rule that is already stored in the lexicon as an inflectional or derivational property, at the cost of introducing a single new primitive. Thanks to this, NooJ can deploy the same resource both to generate inflected forms and to analyse running

text and corpora, without any additional recomposition step.

To support this approach, we created a new swap operator (<W>) for the description of Mwotlap, dedicated to local reordering and allowing adjacent segments to be swapped when required by phonotactic or morphophonological constraints.

3 The Mwotlap language

Mwotlap (endonym *na-vap to-Mōtlap*, ISO 639-3: mlv, Glottocode: motl1237) is spoken by an estimated 2100 people, primarily on Mota Lava and Vanua Lava, in northern Vanuatu. It is the most vibrant language within the Torres-Banks linkage of Oceanic languages, a subgroup of the Austronesian family (François, 2012). Because Mwotlap is spoken by all generations and intergenerational transmission has – so far – run uninterrupted, UNESCO (2025) lists it as a “Safe / Not Endangered” entry in its atlas of endangered languages; however, the *Endangered Languages Project* (ELP, n.d.) recognizes it as “threatened”.

François produced a vast body of research to document the Mwotlap language, beginning in 1997. First and foremost, his doctoral dissertation (François, 2001) constitutes the first complete description of Mwotlap ever written. Before that date, the only publications about this language were a grammar sketch (Codrington, 1885, pp. 310-322), an article about its phonology (Kasarhérou, 1962), and a wordlist (Tryon, 1976).

3.1 Prefixation in Mwotlap

The language distinguishes two types of productive prefixes: invariable prefixes, whose form is constant regardless of the phonological shape of the radical, and copiable prefixes, whose initial vowel slot is sensitive to the radical’s vowel. Table 1 lists the eight copiable prefixes of Mwotlap and the Parts of Speech (PoS) they can be associated to (after François, 2001, p. 100).

Mwotlap has seven phonological vowels: /i, ɪ, ε, a, ɔ, u, u/, spelled respectively as ⟨i, ē, e, a, o, ō, u⟩. By convention, the 8 copiable prefixes (first column of Table 1) are noted with a capitalized vowel: this does not denote an archiphoneme, but rather flags the *default* realization of the prefix in the absence of copy, or of elision (François 2000). Five distinct processes govern the surface form of

Process	Radical	Unprefixed	Prefixed
Copy	QOYQOY	<i>qoyqoy</i>	<i>no-qoyqoy</i>
Insertion	SYIP	<i>siyip</i>	<i>na-syip</i>
Transfer	WUMET	<i>wumet</i>	<i>nu-wumet</i>
Elision	EP	<i>ep</i>	<i>n-ep</i>
Blockage	NEY	<i>ney</i>	<i>na-ney</i>

Table 2: The five vowel rules at the prefix-radical boundary of words exemplified with the addition of the article *na*-

this vowel at the prefix-radical boundary (François, 2005, p. 118), as shown in Table 2.

The first three vowel rules were presented and illustrated in Section 1. The last rule, namely *vowel blockage*, concerns several #CV- roots that do not undergo vowel copy nor vowel transfer, and instead take the default form of copiable prefixes shown in Table 1. For example, if *ney* ‘almond’ were a regular C-initial root, we would expect a prefixed form **ne-ney*, but we observe *na-ney* instead. While this behavior is expected for #CC-roots like SYIP (see 3a), its occurrence with #C-radicals is exceptional, and historically vestigial (François, 2000). To represent these cases on NooJ, it is sufficient to move to the left boundary of a word and attach the default realization of a prefix. As for #V- roots, they do not need any special term in NooJ: the corresponding automation moves the cursor to the left boundary of the word form and directly adds the thematic consonant C₀ of the prefix in that position, with no vowel.

As mentioned before, #CC- roots simply trigger the default realization of the prefix. For these forms, the rule to be applied does not concern the prefixed form, but the unprefixed one (see Table 1), as this is when they require vowel insertion: the epenthetic vowel V breaking #CC- into #CVC- inherits the quality of the vowel in the following syllable, and this is where a new operator for non-concatenative morphology is needed.

4 Formalizing morphology with NooJ

To describe the morphology of a language, NooJ proposes two formal tools: inflectional and derivational grammars (described in .nof files), and morphological grammars (described in .nom files) used to describe other phenomena such as agglutinations (e.g., to analyze word forms such as *antidisestablishmentarianism* and productive neologisms like in “The *retrumpization* of the US policy”). Both systems are represented by

Operator	Function
<L>*	Move cursor left
<R>*	Move cursor right
<E>	Add empty string (ε)
<C>	Change case
<D>	Duplicate
*	Delete preceding character
<S>	Delete next character
<N>*	Go to end of subsequent word
<P>*	Go to end of previous word

Table 3: Generic operators in NooJ .nof ECFGs (Silberstein, 2026). Operators marked with “*” accept arguments.

Enhanced Context-Free Grammars (ECFGs). As inflectional and derivational phenomena are not recursive, they are transparently converted into efficient Finite-State Transducers (FSTs) by NooJ. Linguists therefore can write very expressive and natural grammars while not losing any efficiency during parsing, which allows them to parse large corpora of texts in real time.

Inflectional and derivational paradigms are described by grammar rules that represent sets of affixes that must be added to each lexical entry to produce their corresponding inflected and derived forms, associated with linguistic properties described in the property definition file (.def file). For example, (6) represents the PEN inflectional paradigm in English.

(6) PEN = <E>/+Singular | s/+Plural ;

This rule states that by adding the empty string (ε, encoded as <E> in NooJ) to a lexical entry, one produces its singular form, while adding -s forms the plural. Affixes may contain orthographic operators that can perform special operations at the cursor position. For instance, the operator deletes the current character. To inflect the verb *to eat*, one can use the paradigm in (7).

(7) EAT = <E>/+Base | s/+3+Singular | en/+PastParticiple | ing/+Gerund | <B3>ate/+Preterite ;

In this paradigm, <B3>ate deletes the last 3 characters of the lexical entry, and then adds *ate*: *eat* + <B3>ate → *ate*. The resulting form is associated with property +Preterite.

Although the operator allows the system to produce any wordform from any lexical entry, it is not satisfying in practice as, in certain cases, it

would force linguists to use multiple grammar rules to formalize a single morphological paradigm.

NooJ offers several other operators, listed in Table 3. , <L>, <N>, <P>, <R>, <S> accept arguments to perform their assigned operation more than once, or even until reaching the edge of the current string (e.g. <L2> ‘move left twice’, <LW> ‘move to the left boundary of this word’). On top of these generic operators, operators to describe phenomena specific to many languages were developed, such as <Q> for reduplication in Asian and Amerindian languages, <F> for consonant finalization in Semitic languages, <I> to add a iota to the current Greek letter, etc. To describe Mwotlap morphology, the new operator <W> (swap letters) was added.

The approach presented here relies on a fundamental assumption worth making explicit: that the orthography of the language being formalized is phonetically transparent, meaning that there is a close (ideally one-to-one) correspondence between graphemes and phonological segments, as is the case for Mwotlap (François, 2001, p. 51). A digraph representing a single segment, a diacritic encoded as a separate code point, or a character sequence whose phonological value is context-dependent would all silently shift the cursor to the wrong position, producing incorrect forms without raising an error.

This limitation was evident in the development of the Mwotlap module: the letters ⟨m̃⟩ and ⟨ñ⟩, which represent /ɲm̃w/ and /ɲ/, respectively, have no single precomposed Unicode code point and are encoded as a base character followed by a combining macron (U+0304). A specific NooJ update (v7.3, update b20250410) was required to ensure that each base letter and its diacritic were treated as a single unit by the morphological engine. This is analogous to the handling of Hungarian digraphs such as ⟨sz⟩ and underscores the need for linguists to verify that every grapheme in their language is correctly processed by NooJ. For languages with rich diacritic systems, non-Latin scripts, or orthographic conventions that routinely encode single phonemes as multi-character sequences, additional preprocessing steps of this kind should be anticipated in the module development pipeline.

5 Introducing the Mwotlap operator <W>

In Mwotlap, vowels may be swapped with prefixes under certain conditions – as in the rule of vowel transfer – e.g., (na-) + *wumet* → *nuwmet*. To describe all potential swapping operations with NooJ generic operators, we would need to implement 8 prefixes × 7 vowels, i.e., 56 grammar rules. While technically functional, that is not acceptable from a linguistic point of view, as it would obscure the unitary nature of the process by treating what is, in effect, a single productive rule as 56 idiosyncratic rules. In particular, this would ignore François’ (1999, 2000, 2001) description of vowel behavior in Mwotlap and regress to the inaccurate generalizations of Kasarhérou (1962) and Crowley (2002).

As anticipated, the prefixation of #CCV- radicals and blocking #CV- radicals was easily described as the juxtaposition of two strings of text (namely, the radical and the prefix). What was left to do was computationally describing all the rest, i.e., the prefixation of #CV- radicals, vowel insertion in unprefixated #CCV- radicals, and vowel transfer in prefixed #CV- radicals that undergo it. To do so, we added to NooJ the operator <W> to swap characters in the right position when performing vowel copy, vowel insertion, and vowel transfer.

5.1 Vowel Copy

The rule (8) for vowel copy works such that, given a #C₁V₁ radical, V₁ is taken as the vowel of the CV- prefix to be put before it (François, 2001, p. 100):

$$(8) \quad C_0V_0- + \#C_1V_1C_2- \rightarrow \#C_0V_1-C_1V_1C_2-$$

The corresponding implementation in NooJ moves the cursor to the left boundary of the word form (<LW>), then moves it right twice (<R2>). At this point, the cursor is positioned after the vowel subject to copy (V₁) and, therefore, such a vowel must be duplicated (<D>). To move the copied vowel to the beginning of the word form, we move the cursor left once (<L>) and then swap the two preceding characters (<W>). We then return to the left boundary of the word form (<LW>) and insert the thematic consonant C₀. The resulting word form is the correct prefixed form of the starting word form, now morphologically connotated with

the trait that C_0 brought. The final result is the fully inflected form. For example, *qoyqoy* ‘dream’ is prefixable as *no-qoyqoy* (‘the/a dream’) since the vowel of a radical that undergoes vowel copy is copied into the prefix, and the process is described as (9) in NooJ.

$$(9) \quad qoyqoy\langle LW\rangle\langle R2\rangle\langle D\rangle\langle L\rangle\langle W\rangle\langle LW\rangle n$$

5.2 Vowel Insertion

The strict (C)V(C)|(C)V(C) syllabic structure of Mwotlap, which does not allow syllables of CCV type to exist in the language, was anticipated in Section 3.1. This is why #CCV- radicals, when appearing alone, show an epenthetic vowel that breaks the two consonants with a necessary syllable boundary. For example, *siyip* ‘waterfall’ is SYIP as a radical, prefixable as *na-siyip* ‘the/a waterfall’, and usable as an unprefixable word as *siyip*. The inserted vowel of a radical that undergoes vowel insertion is in fact not part of the underlying radical, but added via such a process.

The formal rule (10) for vowel insertion makes sure that, when appearing without any prefix, a wordform with radical # $C_1C_2V_1$ -, has V_1 inserted between the two initial consonants (François, 2001, p. 79):

$$(10) \quad \#C_1C_2V_1- \rightarrow \#C_1V_1C_2V_1-$$

Among all the formal rules shown in this section, this is the only one that points at unprefixable (“bare”) forms as its objective, starting from the radical contained in a compiled NooJ dictionary (.nod file). The corresponding grammar rule moves the cursor to the left boundary of the word form ($\langle LW \rangle$), then moves it to the right three times ($\langle R3 \rangle$). At this point, the cursor is positioned after V_1 , that must be inserted between C_1 and C_2 . To do so, the vowel is duplicated ($\langle D \rangle$) and, as the cursor moves left once ($\langle L \rangle$), it is then swapped with the preceding character ($\langle W \rangle$). The resulting form is the correct unprefixable form of the starting word form, obtained in NooJ as in (11).

$$(11) \quad siyip\langle LW\rangle\langle R3\rangle\langle D\rangle\langle L\rangle\langle W\rangle$$

5.3 Vowel Transfer

The vowel of a prefix is truly found to be “intermittent” (François, 2000; 2001) in those lexemes that are not easily categorized as either

“copiable” or “blocking” in the sense of vocalic copy, and undergo vowel transfer instead. For example, *wumet* ‘bowl’ is prefixable as *nu-wumet* ‘the/a bowl’, with *nA-* taking on the radical’s /u/, and usable as an unprefixable word as *wumet*. The transferred vowel of a radical that undergoes vowel transfer, is in fact part of the radical and transferred to the prefix via vowel transfer, which follows the rule in (12) (François, 2001, p. 115):

$$(12) \quad C_0V_0- + \#C_1V_1C_2V_2- \rightarrow \#C_0V_1-C_1C_2V_2-$$

The corresponding grammar rule moves the cursor to the left boundary of the word form ($\langle LW \rangle$), then moves it to the right twice ($\langle R2 \rangle$). The cursor is then positioned after the vowel (V_1) that must be moved to the prefix; it is duplicated ($\langle D \rangle$), then moves left once ($\langle L \rangle$), and is swapped with the preceding character ($\langle W \rangle$). The transferred vowel is now in its correct position, and the cursor can now delete it ($\langle S \rangle$). At this point, one can move the cursor to the left boundary of the wordform ($\langle LW \rangle$) and then insert the thematic consonant C_0 . The resulting form is the correct prefixed form corresponding to the starting lexical entry, associated with the linguistic property for C_0 . In NooJ, this is obtained by what is exemplified in (13):

$$(13) \quad wumet\langle LW\rangle\langle R2\rangle\langle D\rangle\langle L\rangle\langle W\rangle\langle S\rangle\langle LW\rangle n$$

6 Evaluation

The NooJ Mwotlap module comes with separate dictionaries for grammatical and lexical morphemes (Meroni, 2026, p. 5). The latter is LWOTLAP.nod, a 2241-entry electronic dictionary in the form of a lexicon-grammar table, containing all monomorphemic lexical roots. It was automatically extracted from François (2026) using a Python pipeline, with particular attention to the conventions it uses: e.g. in a headword, [V] stands for vowel insertion, $\langle V \rangle$ stands for vowel transfer, and a preposed $\circ$ stands for vowel blockage; all other consonant-initial headwords are treated as undergoing vowel copy, while all other vowel-initial headwords are treated as undergoing vowel elision. In LWOTLAP.nod sit 1199 nouns, 602 verbs, 155 adjectives, 19 prefixable locatives and the 12 basic numerals (the terms for designating the numbers 1-10, 100 and

```

bem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+BARE
bem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+AO
nibem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+AO+3+s
malbem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+IAM
etbem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+NEG+REALIS
titbem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+NEG+IRREALIS
mebem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+PFT
tebem,bem,V+Trans+V2+EN="carry"+FR="porter"+FLX=VC_PRED+FUT
ben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+BARE
ben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+AO
niben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+AO+3+s
malben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+IAM
etben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+NEG+REALIS
titben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+NEG+IRREALIS
meben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+PFT
teben,ben,V+Trans+NOVNIInani+EN="light"+FR="allumer"+FLX=VC_PRED+FUT
benwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+BARE
malbenwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+IAM
etbenwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+NEG+REALIS
titbenwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+NEG+IRREALIS
nebenwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+ART
nebenwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+STA
mebenwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+PFT
tebenwar,benwar,N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+FUT
benwar,benwar,+INITIALCATEGORY=N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>
lebenwar,benwar,LOC+INITIALCATEGORY=N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+NounDerived
tebenwar,benwar,ADJ+INITIALCATEGORY=N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+NounDerived
telebenwar,benwar,ADV+INITIALCATEGORY=N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+LOC
bebenwar,benwar,ADV+INITIALCATEGORY=N+Weak+NHum+Inani+Conc+Flora+EN="benoa"+FR="boisnoir"+FLX=VC_N+DRV=VC:<E>+NounDerived
bep,bep,N+Weak+NHum+Inani+Conc+Flora+EN="chilli"+FR="piment"+FLX=VC_N+DRV=VC:<E>+BARE
malbep,bep,N+Weak+NHum+Inani+Conc+Flora+EN="chilli"+FR="piment"+FLX=VC_N+DRV=VC:<E>+IAM
etbep,bep,N+Weak+NHum+Inani+Conc+Flora+EN="chilli"+FR="piment"+FLX=VC_N+DRV=VC:<E>+NEG+REALIS

```

Figure 1: Snippet of inflected forms of LWOTLAP.nod, the main lexicon-grammar of lexical morphemes in the NooJ Mwotlap module.

1000, then composable in all others), all susceptible of being prefixed (see Table 1).

By using the “Inflect” option of NooJ’s “Lab” menu on LWOTLAP.nod, 44380 inflected forms are obtained. Among these, all of the PoS susceptible of being prefixed are correctly produced by the engine, which applies the inflectional and derivational grammars associated with each entry (see Figure 1). This confirms that the implementation successfully captures the formal behavior of the prefix-root boundary while remaining fully productive across the lexicon. Importantly, this approach avoids having to resort to separate lexicalized rules for individual vowel qualities, as was anticipated in Section 2: across the lexicon, the operator correctly derives forms involving local vowel displacement or duplication; and it does so while preserving the phonotactic constraints of Mwotlap, especially the language’s strict (C)V(C) syllable structure. Besides, the implementation of <W> makes it possible to generate forms that would otherwise require extensive rule duplication, thereby reducing coding redundancy while preserving descriptive accuracy. From a computational perspective, this confirms that local segmental reordering can be modeled efficiently within NooJ’s finite-state architecture, without moving outside a tractable formal framework.

A demo text, *Mayanag t-Aliwan.not* (346 types, 1711 tokens; TTR = 0.202), also comes with the NooJ Mwotlap module. Performing a linguistic

analysis on this text, thus applying all the linguistic resources present in the module, will yield all correct analyses of every prefixable PoS, confirming 100% recall and 100% precision – every token receives at least one analysis, and every analysis set contains the correct reading.

The NooJ ambiguity log returns 285 form types with two or more analyses. Of these, 131 are false positives since they carry exactly one lexical or morphological analysis accompanied by one or more syntactic annotations, which do not constitute competing morphological readings. The remaining 154 types (44.5% of the type inventory) are genuinely ambiguous. Their ambiguity arises from well-defined sources unrelated to the vowel rules formalized in this paper. One of them is accidental homophony – e.g. *et* can be a noun ‘person’, a verb ‘to see’, or a clitic of negation. Another cause for ambiguity is lexical flexibility, i.e. the ability, for a single lexeme, to fill different grammatical functions (see François, 2017): e.g. the lexeme *bah* ‘finish+’ may adopt the grammatical behavior of an intransitive verb, of a transitive verb, of an adverb, and of a TAM marker. While these sources of ambiguity reflect the linguistic profile of Oceanic languages more generally, they also combine with some idiosyncrasies of the Mwotlap language: e.g., the often identical surface realization – due to Vowel Copy [§5.1] – of the noun article *nA-*, the stative *nE-* and/or of the 3rd person singular aorist prefix *ni-*. Table 4 addresses this evaluation quantitatively.

Recognition performance

Unique wordforms	346
Recall	346 (100%)
Precision	346 (100%)

Ambiguity profile

Unambiguous	192 (55.5%)
Ambiguous (true positives)	154 (44.5%)
Total	346

Table 4: Evaluation of the NooJ Mwotlap module on *Mayanag t-Aliwan.not* (1711 tokens, 346 types): recall, precision, and ambiguity by source.

Disambiguation of these readings requires syntactic context that the current module does not yet have access to. For the scope of this study, this natural language-based evaluation confirms not only the correctness of the generated forms, but also the ability of the analyzer to recognize and parse naturally occurring prefixed word forms in running text. At the current stage of development, the module prioritizes maximal recall at the morphological level, returning all linguistically valid analyses for a given form.

Developing syntactic disambiguation grammars (.nog files), which are needed to filter competing analyses given their surrounding context, is a primary direction for future work.

7 Concluding remarks

Thanks to the creation of the new <W> operator, three superficially distinct linguistic processes can now be formalized as a single unified paradigm. In the case proposed here, vocalism in Mwotlap has a unitary sensitivity to the radical's phonological shape, not an arbitrary list of vowel-specific correspondences.

The three morphophonological processes described in this study are not specific to Mwotlap, as analogous phenomena are attested across typologically and genealogically unrelated languages (see Section 1). In all cases covered, the shared structural property is the same, and the new <W> operator (perform a swap of two adjacent letters), allows linguists to unify their formalization. It is therefore reasonable to expect that any language where morphophonological generativity involves local segment reordering, vowel spreading across a consonant boundary, or epenthesis conditioned by a following nucleus, could benefit from this operator, or a variant of it. Formalizing such phenomena would no longer require linguists to enumerate separate rules for each vowel quality, as

the operator abstracts over segmental identity entirely. Future work should test this expectation against other languages.

Glossing conventions

1PL	1 st person plural	ART	article
ANT	anterior	FV	final vowel
APPL	applicative	SUBJ	subject

References

- Saba Amsalu and Girma A. Demeke. 2006. Non-concatenative finite-state morphotactics of Amharic simple verbs. *ELRC Working Papers*, 2(3): 304-325.
- Peter Barclay. 2026. A Rule-Based Computational Model for Gàidhlig Morphology. In *Proceedings of the 18th International Conference on Agents and Artificial Intelligence*, Volume 1. Scitepress, pages 683-693.
<https://doi.org/10.5220/0014654900004052>
- Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford, CA (distributed by the University of Chicago Press).
- Lyle Campbell. 1974. Theoretical implications of Kekchi phonology. *International Journal of American Linguistics*, 40(1): 269-278.
<https://www.jstor.org/stable/1265001>
- Noam Chomsky and Marcel P. Schützenberger. 1963. The algebraic theory of context free languages. In *Computer Programming and Formal Systems*. North Holland, pages 118-161.
- Robert H. Codrington. 1885. *The Melanesian Languages*. Clarendon Press, Oxford, United Kingdom.
- Yael Cohen-Sygal and Shuly Wintner. 2006. Finite-State Registered Automata for Non-Concatenative Morphology. *Computational Linguistics*, 32(1):49–82. <https://doi.org/10.1162/coli.2006.32.1.49>
- Terry Crowley. 2002. Mwotlap. In *The Oceanic Languages*. Curzon Press, pages 587-598.
- ELP. n.d. *Motlav*.
<https://endangeredlanguages.com/elp-language/motlav>
- Axel Fleisch. 2000. *Lucazi: A Morphosemantic Analysis*. Köppe, Köln, Germany.
- Alexandre François. 1999. Mouvements et clonages de voyelles en motlav: entre phonologie et morphologie. *Bulletin de la Société de Linguistique*, 94(1): 437-486.

- Alexandre François. 2000. Vowel shifting and cloning in Motlav: historical explanation vs. formal description. In *Proceedings of AFLA 7 (The Seventh Meeting of Austronesian Formal Linguistics Association)*. Vrije Universiteit Amsterdam, pages 49-68.
- Alexandre François. 2001. *Contraintes de structures et liberté dans l'organisation du discours. Une description du mwotlap, langue océanienne du Vanuatu*. Doctoral Dissertation. Université Paris-IV Sorbonne, Paris, France.
- Alexandre François. 2005. A typological overview of Mwotlap, an Oceanic language of Vanuatu. *Linguistic Typology*, 9(1): 115-146. <https://doi.org/10.1515/lity.2005.9.1.115>
- Alexandre François. 2012. The dynamics of linguistic diversity: Egalitarian Multilingualism and Power Imbalance among Northern Vanuatu Languages. *International Journal of the Sociology of Language*, 38(214): 85-110. <https://doi.org/10.1515/ijsl-2012-0022>
- Alexandre François. 2017. The Economy of Word Classes in Hiw, Vanuatu: Grammatically Flexible, Lexically Rigid. *Studies in Language*, 41(2): 294-357. <https://doi.org/10.1075/sl.41.2.03fra>
- Alexandre François. 2026. *A Mwotlap–English–French cultural dictionary*. CNRS, Paris, France. <https://tiny.cc/Mwotlap-dict>
- Nancy Hall. 2006. Cross-linguistic patterns of vowel intrusion. *Phonology*, 23(3): 387-429. <https://doi.org/10.1017/S0952675706000996>
- Mans Hulden. 2009. Foma: a Finite-State Compiler and Library. In *Proceedings of the Demonstrations Session at EACL 2009*. Association for Computational Linguistics, pages 29-32.
- Jacqueline Kasarhérou. 1962. Les changements vocaux de trois préfixes en motlav. *Te reo*, 5(1): 32-34.
- Martin Kay. 1987. Nonconcatenative Finite-State Morphology. In *Third Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 2-10.
- Kimmo Koskenniemi. 1983. Two-level Model for Morphological Analysis. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Volume 2. BRD, pages 683-685.
- Paul R. Kroeger. 2005. *Analyzing Grammar: An Introduction*. Cambridge University Press, Cambridge, United Kingdom.
- Lutz Marten and Nancy Chongo Kula. 2002. Semantic transparency in phonology: telicity and vowel copying in Deiriku. In *Phonologica, Proceedings of the 20th International Phonology Meeting*. Cambridge University Press, pages 1-17.
- Fabio Meroni. 2026. [Introducing Mwotlap Language Resources for NooJ](#). In *Formalizing Natural Languages: Applications to Natural Language Processing and Digital Humanities. 19th International Conference, NooJ 2025, Paris, France, June 11–13, 2025, Revised Selected Papers*. Springer, pages 3-15. https://doi.org/10.1007/978-3-032-17103-0_1
- Kenneth Olson. 2001. *The phonology and morphology of Mono*. Doctoral Dissertation. University of Chicago, Chicago, IL.
- Femphy Pisceldo, Rahmad Mahendra, Ruli Manurung, and I Wayan Arka. 2008. A Two-Level Morphological Analyser for the Indonesian Language. In *Proceedings of the Australasian Language Technology Association Workshop 2008*. Association for Computational Linguistics, pages 142-150.
- Max Silberztein. 2016. *Formalizing Natural Languages: the NooJ Approach*. Wiley, Hoboken, NJ.
- Max Silberztein. 2024. [Preface](#). In *Linguistic Resources for Natural Language Processing*. Springer Nature, pages xviii-xvi. <https://doi.org/10.1007/978-3-031-43811-0>
- Max Silberztein. 2026. *NooJ v7 Manual*. Université Marie et Louis Pasteur, Besançon, France. <https://nooj.univ-fcomte.fr/tutorials.html>
- Harald Trost. 1990. The application of two-level morphology to non-concatenative German morphology. In *COLING 1990*, Volume 2. Association for Computational Linguistics, pages 371-376.
- Darrell T. Tryon. 1976. *New Hebrides Languages: An internal classification*. Pacific Linguistics, Canberra, Australia.
- UNESCO. 2025. *World Atlas of Languages*. <https://en.wal.unesco.org/>
- Shouly Wintner. 2014. [Morphological Processing of Semitic Languages](#). In *Natural Language Processing of Semitic Languages. Theory and Applications of Natural Language Processing*. Springer, pages 43-66. https://doi.org/10.1007/978-3-642-45358-8_2