A Morphological Transducer for the Limbu Language

Avyaya Singh, Jonathan North Washington

Swarthmore College
500 College Avenue, Swarthmore, PA 19081

{asingh5, jwashin1}@swarthmore.edu

Abstract

This paper presents the first ever morphological transducer for the Limbu language, also known by its endonym Yakthung Pan, an endangered Sino-Tibetan language primarily spoken in the area known as Limbuwan/Koshi Province in Eastern Nepal, with a minority population in the Sikkim state of India, where Limbu enjoys official status. Using a corpus of Limbu text produced by field interviews (Michailovsky, 1977), and a translation of the Holy Bible into the Limbu language, the paper presents various elements of the morphology of the language, how they were implemented into the transducer, and an evaluation of the transducer against identified corpora and a gold standard. With a relatively small lexicon, the transducer was found to have reasonable coverage (~60%), with high precision (88%) but low recall (32%). The paper discusses future expansion through further involvement with the community, which can help in the maintenance and revitalisation of the endangered language.

Keywords: Limbu, morphological transducers, morphological analysis, Tibeto-Burman languages, South Asia, Nepal

1. Introduction

This paper describes the creation of the first-ever morphological transducer for the Limbu language of Eastern Nepal, using tools from Apertium, a free and open source digital platform intended primarily for rule-based machine translation.

Linguistic preservation and revitalisation of Limbu is currently being undertaken by groups such as Yakthung Cho, an online Limbu Art collective, run by Subhas Thebe Limbu, a Limbu-origin film director, which seeks to produce Limbu language learning tools through online classes and produce content in the Limbu language aimed at informing the Limbu audience of issues pertaining primarily to their efforts at language revitalisation and autonomist demands in Eastern Nepal.[1] The first author of this paper themselves joined such a class conducted by Yakthung Cho in the English medium to learn the Limbu language.

A morphological transducer, here implemented as a finite-state transducer (FST), both analyses text in the language, providing tokenisation of the text and a morphological analysis of each token (morphological analysis), and creates tokens from input analyses (morphological generation). Morphological transducers can prove helpful in learning vocabulary (Katinskaia et al., 2018) and morphology (Arppe et al., 2022); as elements in word-form creators (Fernald et al., 2016; Kazantseva et al., 2018); and as the basis for spell checkers (Washington et al., 2021). They can further also be used in rule-based and hybrid machine translation systems (Khanna et al., 2021), and in a range of other NLP tasks and pipelines.

This paper is structured as follows. Section 2 provides background into the Limbu language and its general structure. Section 3 describes the methodology of creating the transducer, and has details on encountered challenges. Section 4 presents an evaluation of the transducer, showing its coverage over various corpora as well as precision and recall over a hand-annotated gold standard. Section 5 concludes and discusses future work. Associated tools, including a keyboard and a disambiguator, are also described.

2. Background on Limbu/Yakthung

Limbu, known by its endonym Yakthung Pan (ᤕᤠᤰᤌᤢᤱ ᤐᤠᤴ), is spoken by 350,436 people in Nepal (National Statistics Office, 2025), and by 40,835 speakers in North-Eastern India (Office of the Registrar General and Census Commissioner, 2018), mostly belonging to the Limbu ethnicity.[2] Figure 1 shows the geographic distribution of speakers.

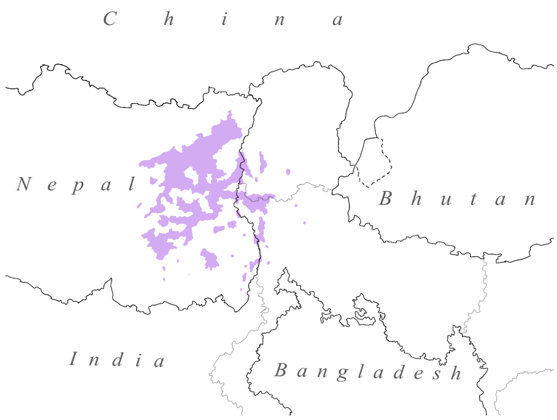

Figure 1 : A map highlighting the regions in which the Limbu language is spoken, namely, Eastern Nepal and the Sikkim state of India.[3]

[1] https://www.instagram.com/yakthungcho/?hl=en

[2] https://www.ethnologue.com/language/lif/

[3] https://commons.wikimedia.org/wiki/File:Limbu.map.png

Limbu belongs to the Kiranti subfamily of the Tibeto-Burman languages within the wider Sino-Tibetan language family. The Kiranti subfamily, spoken in Eastern Nepal includes related languages such as Sunuwar, Rai, and Yakkha. Limbu is spoken primarily in the region known to the Limbu people as Limbuwan, called the Koshi Province by the Nepalese government, and it enjoys official status in the province.[4] Although the province is autonomous, due to the federalisation of the Nepalese government following the republican revolution of 2006,[5] the Limbu people have been demanding the reinstatement of Limbuwan through the renaming of the Koshi province, as these lands are seen as ancestral to them.[6] The language is also spoken by a minority in the Indian state of Sikkim which borders this province to the east, where it also enjoys official status.

Limbu is a highly polysynthetic language: words are inflected using a series of affixes, and multiple words can fuse together. The language has four main dialects: Phedape, Chhathare, Tambarkhole, and Panthare (van Driem, 1987). Interesting linguistic features include dual number, split ergativity, and double (sometimes triple) negation, and primarily Subject-Object-Verb word order. Limbu uses its own indigenous orthography, called the Sirijonga script (§3.3.1).[7]

The phonemic inventory of Limbu is very similar to those of nearby Indo-Aryan languages, although it is itself Sino-Tibetan; this is because it is part of the Nepali Sprachbund. Thus, its phonological inventory and orthography follow the Indo-Aryan *Varga* system, where consonants are divided into 5 *Vargas* according to place of articulation. Thus, in terms of its phoneme inventory, the language and its orthography are comparable to Nepali, Bengali, Hindi, etc. (van Driem, 1987).

Only two reference grammars on the language were available to the authors, both primarily concerning the aforementioned Chhathare dialect: van Driem (1987) and Tumbahang (2007). These grammars were consulted in the creation of the transducer. Because of the more specific dialectal focus of Tumbahang (2007), words/phrases encountered in this source were included as synonyms/alternate spellings of words found in van Driem (2007), in an effort to make the transducer dialect-aware but generalisable for the Limbu language as a whole.

[4] https://www.ethnologue.com/language/lif/

[5] https://interactive.aljazeera.com/aje/2016/nepal-maoist-dream/index.html

[6] https://kathmandupost.com/province-no-1/2023/03/12/general-strike-throws-life-out-of-gear-in-koshi-province

[7] https://www.ethnologue.com/language/lif/

The authors do not speak the Limbu language, but this transducer was created through close study of the aforementioned reference grammars and deductions made through use of linguistic analysis. Future work will include native speakers.

3. Methodology & Implementation

This section goes over the details of the transducer, namely how it was constructed (§3.1), how its lexicon was expanded (§3.2), and solutions to challenges the authors encountered while creating the transducer (§3.3).

3.1 Tools used

The transducer described was created entirely by hand and is freely and publicly available under the GPL v3 Free/Open Source license.[8] The intent is to increase access, as no other transducer is known to exist for the Limbu language, and resources on the language are scarce. The transducer is also available online for morphological analysis and generation via Apertium's web UI.[9] Machine learning approaches were not used, due to their need for large corpora and multiple development cycles, the difficulty in correcting errors, and their lack of precision (Butt, 2021).

The lexd formalism (Swanson and Howell, 2021) was used to implement the morphotactics of Limbu. Previously published lexd transducers showcase its ability to handle non-suffixational morphology (Washington et al., 2021; Christopherson, 2023) and some discontiguous morphology (Swanson and Tyers, 2022), both of which are used extensively in Limbu. The transducer uses the Apertium framework (Forcada et al., 2011; Khanna et al., 2021) for compilation scripts and the HFST format and tools (Lindén et al., 2011) for storing and working with the compiled transducer. Apertium also offers a handful of tools for leveraging a morphological transducer for language maintenance, learning, and revitalisation, which are all future directions the authors intended the transducer for.

The morphophonology was implemented in the twol formalism, included with HFST. The analyser and generator are compiled by compose-intersecting the morphotactics transducer and the morphophonology transducer, as in sources cited above for other lexd transducers.

3.2 Lexicon

The transducer was developed against the only two available digital corpora of Limbu text of any substantial size: the Wycliffe translation of the Bible into the Limbu language (Wycliffe, 2009) and a collection of field interviews in the Limbu language compiled primarily by linguists Boyd

[8] https://github.com/apertium/apertium-lif.git

[9] https://beta.apertium.org/#analysis?aLang=lif

Michailovsky and Martine Mazaudon in 1977 (Michailovsky, 1977), available on the Pangloss website.[10,11] This latter source is not in the official Indigenous script of the language, and thus parts of it were hand-transliterated by the first author for the purposes of evaluating this transducer. The first author used a dictionary app in Limbu, available online, which contains basic words from every dialect, as a primary source for specific words in the language.[12] Besides these primary sources, the first author also used an online database[13] and dictionary[14] of Limbu words, both presented in the Latin transcription.

Several methods were used to expand the lexicon. One involved examining the most frequently used words in the Limbu Bible translation and cross-referencing them with an English translation. Through this method, many verbs and proper nouns were able to be added (such as God, David, Solomon, Earth, Heaven; saying, doing, commanding). A similar method was used in relation to parts of the Pangloss corpus (Michailovsky, 1977), focussing on the *Pandy Dancing - A Conversation* interview,[15] one of the longest of the collection of interviews in this source. Further, the first author used the aforementioned Dictionary app and reference grammars to check noun stems and other parts of speech. Through an iterative development cycle (Tyers et. al, 2015) using these resources, the first author was able to generate a basic lexicon for Limbu, made up of the most common words encountered in the corpora (Table 1).

Parts of Speech	Tag(s)	Stems
Numerals	<num>	157
Nouns	<n>	148
Proper Nouns	<np>	131
Pronouns	<prn>	56
Verbs	<v>	51
Adverbs	<adv>	38
Ordinals	<num><ord>	22
Adjectives	<adj>	17
Prepositions	<pr>	10
Conjunctions	<cnj>	10
Determiners	<det>	7
Interjections	<ij>	5
Others		52
Total		**704**

Table 1: Lexicon size by part of speech

[10] https://pangloss.cnrs.fr/corpus/Limbu

[11] These sources are not entirely ideal, but little else is available. The work was partly informed by the Limbu classes the first author joined. A Limbu Wikipedia (https://incubator.wikimedia.org/wiki/Wp/lif/Main_Page) has been added to Wikimedia Incubator, but as of yet, it only contains a few sentences in the language. A translation of the Universal Declaration of Human Rights also exists, available only as an image.

[12] https://play.google.com/store/apps/details?id=com.jettyland.limbudictionary

[13] https://starlingdb.org/cgi-bin/query.cgi?basename=%2fdata%2fsintib%2flimet

[14] https://pangloss.cnrs.fr/dictionaries_content/limbu/dico.html, compiled by Boyd Mikhailovsky

[15] https://doi.org/10.24397/pangloss-0004182

The transducer consists of 125 tags, mostly following the Apertium tagset,[16] defined in the lexd formalism as multi-character symbols, between less-than '<' and greater-than '>' characters. 11 of these tags cover the basic parts of speech as described in Table 1; the rest correspond to subcategories (e.g., noun class and pronoun type), morphological functions (e.g., number, negation marking, person marking, etc.), and miscellaneous POS-level tags (e.g., emphasis and question markers and punctuation tags).

3.3 Challenges

This section discusses a number of challenges encountered in the creation of the transducer: the Yakthung indigenous orthography (§3.3.1), discontinuous verbal morphology (§3.3.2), and morphological ambiguity (§3.3.3).

3.3.1 Orthography

The Limbu language uses an Indigenous orthography called the Sirijonga script, which was purportedly designed by Limbu King Sirijanga Hang in the 9th century, and was then revised by Te-ongsi Sirijunga Xin Thebe in the 18th century, who, due to his work on the Limbu language, came to be considered the reincarnation of the former. No text from before the 18th century exists, and thus, it is believed that Xin Thebe actually designed the script himself (Gaenszle, 2021). An abugida which descends from the wider Brahmi family of scripts used in the region, it is supported by Unicode[17] (original proposal: Michailovsky, 2002). An example of the script is provided in Figure 2.

Figure 2: A sentence in the Sirijonga script of the Limbu language, *yakhtuŋ pan chigemasum*, translating to "You've lost a little of the Limbu language" (Michailovsky, 1977).[18]

Because Sirijonga is an abugida where vowels are diacritics and there are many morphophonological alternations, it was often more straightforward to encode segments as multicharacter non-Sirijonga symbols that could be resolved to Sirijonga based on context. An example of this is the verb 'to do' (Figure 3), encoded entirely with multicharacter symbols. The twol rule in Figure 4 ensures that the {Kk} symbol is realised as ◌ᤴ [k] or ᤺ [g] depending on phonological context. Examples of resulting forms are shown in Figure 5.

Figure 3: Encoding of the verb *chōkma* 'to do' in the lexd formalism using multicharacter symbols.

[16] https://wiki.apertium.org/wiki/List_of_symbols

[17] https://www.unicode.org/charts/PDF/U1900.pdf

[18] *Paddy-dancing - a conversation*, Sentence 292. https://doi.org/10.24397/pangloss-0004182#S292

-3	0	+2	+4	+7	+8	+9	+12
ᤀᤠ- *a-* SUBJ.1PL.INCL-	ᤀᤠᤷ *ab* shoot	**-Ø** -NPST	- ᤢ *-u* -OBJ.3	- ᤶ *-m* -SUBJ.1PL	-ᤛᤡ *-si* -SUBJ.NSG	- ᤶ *-m* -SUBJ.1PL	**-Ø** -PERF
ᤁᤧ- *ke-* SUBJ.2PL-	ᤒᤠ᤹ᤷ *baʔr-* speak	**-Ø** -NPST	- ᤡ *-i* -OBJ/SUBJ.PL		**-Ø** -OBJ.SG		**-Ø** -PERF

Table 2: Morphological breakdown of two example verb forms: (a) ᤀᤠᤀᤠᤒᤢᤶᤛᤡᤶ *aabumsim* 'We (incl) have shot them' and (b) ᤁᤧᤒᤠ᤹ᤖᤡ *kebaʔri* 'You (pl) speak (obj)'. **Ø** indicates an empty slot corresponding to a default.

"{Kk} becomes ᤃ if succeeded by a vowel"
%{Kk%}:ᤃ <=> _ %>: (:ᤢ) (:0) :Vowel ;

Figure 4: Twol rule for the {Kk} symbol.

ᤂᤥᤱᤁᤡ:ᤂᤥᤱᤁᤡ<vbdo><inf>
ᤂᤥᤃᤢ:ᤂᤥᤱᤁᤡ<vbdo><nonpast><s_sg3><o_3><o_sg><perf>

Figure 5: Example forms analysed by the transducer involving the twol rule from Figure 2, with a [k] in the first example a [g] in the second. The ':' separates forms from analyses.

In order to fully implement the morphophonological alternations of the language, 45 twol rules were encoded.

In addition to this project, the first author also created a keyboard for the Sirijonga script, as currently available keyboards were not fully intuitive, as cited by members of the community (p.c., February, 2025). Existing keyboards were based on QWERTY,[19] created by a non-Limbu Nepali artist,[20] or based on the consonant *Varga* system (available in Google's Gboard[21]). The keyboard, developed based on glyph-frequency, was made in consultation with members of the Limbu community, who commented on its effectiveness, and it is available under the MIT License on GitHub as a free download.[22] It was first created using iBus-m17n,[23] and documentation of the layout is also available.[24] In order for seamless usage across platforms, the first author worked with Translations Commons, a language digitisation NGO, on making a mobile-friendly version of the author's produced keyboard. This Free/Open Source keyboard, made using the Keyman system,[25] is supported on iPadOS, Android, iOS, and Windows, and is available for free distribution by the author (60 downloads over the last 2 months) and on the Translations Commons website,[26] as well as on the official Keyman website.

3.3.2 Verbal Morphology

Being a highly polysynthetic language, Limbu can agglutinate up to ~15 suffixes and ~5 prefixes (including cliticised adverbs) to a single verb root, and also incorporate other parts of speech into a verb form (van Driem, 1987). There are three negation affixes that can appear in Limbu also verbal morphology, amounting to the same meaning (the language does not possess double negation as a form of emphasis). For each verb, no affix is mandatory, and thus, a root can exist independently, or with the addition of up to all ~20 affixes. The absence of a filled affixal slot can either mean 'default' for that particular slot (singular for number, first person for person, active for voice, etc.) or simply be an open slot, not providing any additional meaning to the verb. Example verb forms are shown in Table 2, and one pattern for verbal morphology included in the transducer is shown in Figure 6.

Adverbs? + SPN(1) + SPN(4) + Neg?(1) + **Verb-Stems** + Neg?(2) + Reflexivity? + Tense + SPN(5) + SPN(2) + ObjSlot + SPN(6) + NegS? + SPN(7) + ObjNum + SPN(8) + SPN(3) + NegS2? + Mode? + Aspect + Clitic? + Nom? + Subord? + *Adverbs?*

Figure 6: One verbal morphology pattern in the transducer, slightly simplified. *Italics* (added for demonstration) represent lexicons containing foreign roots, **Bold** (added for demonstration) represents the lexicon containing verbal roots, SPN represents the lexicon containing affixes related to the subject's person and number, Neg represents negation lexicons. Numbers in parentheses represent lexicon columns. Some columns include only tags that match prefix morphemes.

Although 13 other such patterns for verbs exist, as verbal morphology varies greatly in the language, the patterns are precise, meaning an error cannot occur in identifying which pattern a verb follows (and thus which slot an affix falls in). Patterns were identified based on van Driem (1987), Michailovsky (1977), and linguistic analysis on forms not encountered in these sources. The morphology of nouns is only somewhat less complex, with 7 patterns being

[19] https://keyman.com/keyboards/sil_limbu_phonetic
[20] https://keyman.com/keyboards/sil_limbu_typewriter
[21] https://support.google.com/gboard/answer/6380730#zippy=%2Cfind-supported-languages
[22] https://github.com/SwatLangTech/Limbu_Keyboard
[23] https://github.com/ibus/ibus-m17n
[24] https://wikis.swarthmore.edu/ling073/Limbu/Keyboard
[25] https://keyman.com/
[26] https://reco.to/JZazvh, https://translationcommons.org/keyboard-creation-projects/published-keyboards/

included. The patterns included so far in the transducer cover the most common morphological patterns in the language.

3.3.3 Ambiguity

The aforementioned 15+5 affixes have many variants, due to phonological alternations, dialectal variation, and the fact that there appear to simply be synonymous morphemes. This is compounded by the fact that there exist 13-15 noun cases (van Driem, 1987) in this language, with each having between 2-5 lexical variants, with the genitive case having the most variants, at 11 (Figure 7), attributed to both orthographic variation and dialectical differences in the language. In the lexd file, the first line of every block of synonymous morphemes is used to denote the ‘standard form’, selected based on the reference grammars. The subsequent variants are either dialectical variants as denoted by Tumbahang (2007) or lexical variants encountered in the Pangloss collection of Limbu interviews. All forms besides the "standard form" include Dir/LR in the comment, which allows them to be included in the analysis transducer but excluded from the generation transducer.[27] Currently, the case lexicon itself contains 51 lines of code in order to account for these variations, 36 of which are considered variants.

```
<gen>:>{NG}{A}{n}
<gen>:>ᤖᤧ                     # Dir/LR
<gen>:>ᤗᤧ                     # Dir/LR
<gen>:>{l}ᤗᤧ                  # Dir/LR
<gen>:>{ie}{l}ᤗᤧ              # Dir/LR
<gen>:>{a}{ie}{l}ᤗᤧ           # Dir/LR
<gen>:>ᤖ᤹                     # Dir/LR
<gen>:>ᤗ᤹                     # Dir/LR
<gen>:>{l}ᤗ᤹                  # Dir/LR
<gen>:>{ie}{l}ᤗ᤹              # Dir/LR
<gen>:>{a}{ie}{l}ᤗ᤹           # Dir/LR
```

Figure 7: Variants of the genitive case based on dialectical or orthographical variation.

Additionally, some morphemes appear to be syncretic with other morphemes, despite being rarely used. For example, -/re/ is one of the alternative genitive case markers, but -/re/ can also mark vocative and ergative. Hence any token containing -/re/ has at least 3 analyses.

A disambiguator was created in VISL CG-3 (Bick and Didriksen, 2015) to reduce the ambiguity of analyses by context. An excerpt is provided in Figure 8—one heuristic which deals with the ambiguity of the genitive suffix, with an example of its application in Figure 9.

```
"<ᤃᤡᤖᤧ>"
        "ᤃᤡ" n sg gen SELECT:85
;       "ᤃᤡ" n sg erg SELECT:85
;       "ᤃᤡ" n sg voc SELECT:85
"<ᤀᤖᤧᤶ>"
        "ᤀᤖᤧᤶ" n sg
        "ᤀᤖᤧᤶ" n ser3sg
;       "ᤀᤖᤧᤶ" n attr REMOVE:293
```

Figure 9: Disambiguation of forms in context (‘wheat-GEN bread’) employing the rule in Figure 8 ("85"). Analyses with preceding ; are removed by the rules from the list of possible analyses.

```
LIST NOUN = n np ;
LIST NGEN = (n gen) (np gen) ;

SELECT NGEN IF
   (0 NGEN)
   (1 NOUN) ;
```

Figure 8: A CG-3 rule from the disambiguator.

The Limbu disambiguator (apertium-lif.lif.rlx) consists of 88 rules. The performance of the disambiguator is discussed in §4.1.

4. Evaluation

For the sake of evaluating this transducer, several corpora were prepared using the previously described only extant digital corpora of Limbu text in the Sirijonga orthography: the Bible corpus. Genesis 1 & 2 as well as 10 sentences (of 324) from the *Paddy Dancing* interview were used for development and are hence treated separately.

4.1 Naive Coverage and Ambiguity

The transducer[28] was evaluated in terms of naïve coverage, which is defined as the number of forms in a corpus which return an analysis, irrespective of whether this analysis is correct or not. The ambiguity of the transducer on these corpora, defined as the average number of analyses returned by the analyser per analysed token, was also calculated with and without the disambiguator.[29] The results are in Table 2.

corpus	tokens	coverage	ambiguity before / after
10 sents	177	100.00 %	3.82 / 1.62
Gen. 1 & 2	1253	63.05 %	3.03 / 1.69
Full Bible	759K	60.62 %	2.28 / 1.44

Table 2: Naive coverage and ambiguity on development and test corpora.

It's seen that coverage is reasonable for a young transducer, at over 60%. Ambiguity is reduced greatly by the disambiguator, from ~3 analyses per token to around 1.5.

4.2 Accuracy

Precision and recall were measured to understand the accuracy of the transducer. Precision is defined as the percentage of analyses returned by the transducer for each form that are correct. Recall is defined as the percentage of the correct analyses for each form that are returned as opposed to not being returned at all.

[27] See Washington et al. (2021, 187) for details on this approach.

[28] The transducer was last evaluated on revision 4b0ebbb in March, 2026.

[29] Scripts used for calculating coverage, ambiguity, precision, and recall are available in the `dev/` directory in the transducer's git repository.

In order to measure accuracy, the first author randomly selected 100 unique tokens (of ~4k) from the *Paddy Dancing* interview, hand-transliterated them, and annotated them manually to create a gold standard. The words in the gold standard appeared to be generally representative of the lexicon and morphology of texts the authors have worked with. The gold standard attempted to capture all possible analyses for each token, by using the glosses provided by Mikhailovsky (1977) and leveraging the first author's knowledge of Limbu morphology. The gold standard was compared to the list of analyses returned by the analyser.

Precision was measured at ~88% (108/122 analyses), while recall was only ~32% (108/340 analyses), with an F-score of 0.4675; i.e., most analyses returned by the transducer were correct (except in two instances, noted below), but many correct analyses were not returned due to the fact that the transducer has low coverage on the interview.

A preliminary qualitative evaluation of forms that did not have an analysis returned by the transducer yields several patterns. Besides missing lexical items (including verb stems, adverbs, nouns, classifiers, etc.), there were a few uncommon morphotactic patterns not implemented in the transducer, mostly for verbs, such as polar question and reduplicated forms. Additionally, there are some mistakes in pattern definitions that cause some patterns not to function fully as intended; e.g., while the language allows up to three negation morphemes in a verb, and the morphotactics were designed to support this, currently the transducer only recognises up to two negation morphemes in a verb form.

Given the fact that the majority of analyses not returned by the transducer were caused by stems missing from the lexicon, the low recall result substantiates the fact that the stems in the lexicon are indeed more common than those missing. Assuming similar numbers of possible analyses per form, and given the random unique sampling approach used to build the gold standard, if included and missing stems were evenly distributed in frequency, then recall would approximate coverage.

There were very few forms with incorrect analyses returned from the transducer; all involve hypothetical but unattested verb forms which are synonymous with another word. For example, the word ᤆᤠᤷ *char* refers to the numeral “4”, however, it is also the predicted verb form involving the optative suffix -ᤷ *-r* added to the verb ᤆᤠᤶ *chama*. However, ᤆᤠᤷ *char* is always used to reference the numeral, and an alternative optative form, ᤆᤠᤸ *chal*, is instead used for ᤆᤠᤶ *chama*.[30] Thus, the optative analyses returned by the transducer in this instance are incorrect. Similarly, the word ᤘᤠ᤹ᤀᤡ *waʔi* refers to a polar question verb form of the stem ᤘᤠᤶ *wama* "to be" in all contexts, rather than any of the other analyses returned by the transducer, despite these being predicted by the morphotactics. One of these analyses involves object marking, which is currently incorrectly allowed for verbs which are not transitive.

[30] For other verbs, the two optative forms are used interchangeably.

5. Conclusion & Future Work

This paper presents the first ever morphological transducer for the endangered Limbu language, to the best of the authors' knowledge. With significant coverage on the Limbu Bible, and with high precision, this paper contributes to the linguistic description of the language, and introduces the ability to perform various text-processing tasks in the language, enhancing the digital presence of the language.

In terms of future work on the language, the authors believe that coverage can be expanded in two main ways: (1) by hand-transcribing Limbu print books, the Universal Declaration of Human Rights, and the interviews from Mikhailovsky (1977), automating transliteration of other sources, and expanding the lexicon based on their contents, and (2) working more closely with native speakers to make sense of forms encountered in these sources that are not able to be understood using published resources. The transducer is easily scalable through adding to the lexicon, tweaking the existing morphological patterns, and adding new ones.

As mentioned before, this transducer can also be used as a component in a rule-based machine translation system. Given that there are currently no MT systems for the Limbu language, the first author did create a basic Limbu-English MT system as a proof-of-concept. However, it is still quite barebones, and is only able to handle a few simple phrase types and words of the language. The authors seek to expand this MT system in the future.

This transducer and its associated projects (e.g., the Limbu keyboard) have already been introduced to the Limbu community. There appears to be some initial uptake, and community members have commented on the efficiency and usefulness of these projects.

Acknowledgments

The authors would like to thank Subhas Thebe Limbu[31,32] and Sumi Limbu[33] for their support and for their help in looking over our work. We would also like to thank all members of the Limbu community for keeping this language alive and thriving, and Yakthung Cho for their contribution

[31] subashthebe.com/

[32] eyebeam.org/artists/subash-thebe-limbu/

[33] scholarworks.uark.edu/etd/5325/

to Limbu linguistic preservation through online classes.[34]

References

Arppe, Antti & Poulin, Jolene & Neitsch, Andrew & Harrigan, Atticus & Hieber, Daniel (2022). itwêwina: Towards a morphologically intelligent and user-friendly online dictionary of Plains Cree. 10.13140/RG.2.2.33073.08804.

Bick, Eckhard & Didriksen, Tino. (2015). CG-3 - Beyond Classical Constraint Grammar. Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015), Vilnius, Lithuania.

Butt, Miriam (2020). Building Resources: Language Comparison and Analysis. At the Second Workshop on Computational Research in Linguistic Typology (SIGTYP 2020), invited to talk. https://youtu.be/D2AzyAY_3Mc.

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

van Driem, George (1987). A Grammar of Limbu, Berlin, New York: De Gruyter Mouton. https://doi.org/10.1515/9783110846812

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

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

Gaenszle, Martin (2021). The Limbu Script and the Production of Religious Books in Nepal. Philological Encounters. 6 (1–2): 43–69. doi:10.1163/24519197-bja10014. ISSN 2451-9197.

Katinskaia, A., Nouri, J., and Yangarber, R. (2018). Revita: a language-learning platform at the intersection of ITS and CALL. In Proceedings of LREC: 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan.

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

Khanna, T., Washington, J., Tyers, Francis M., Bayatlı, Sevilay, Swanson, Daniel G., Pirinen, Tommi A., Tang, Irene, and Alòs i Font, Hèctor (2021). Recent advances in Apertium, a free / open-source rule-based machine translation platform for lowresource languages. Machine Translation.

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

Michailovsky, Boyd; Everson, Michael (2002). L2/02-055: Revised proposal to encode the Limbu script in the UCS (PDF). Retrieved 2026.

Michailovsky, Boyd, Mazaudon Martine (1977). Pangloss Collection | Limbu Corpus. URL: https://pangloss.cnrs.fr/corpus/Limbu?lang=en&mode=norma

National Statistics Office (2025). Languages in Nepal. Kathmandu: National Statistics Office. (National Population and Housing Census 2021). URL: https://censusnepal.cbs.gov.np/results/files/result-folder/Language%20in%20Nepal.pdf

Office of the Registrar General & Census Commissioner (2018). Language: India, States and Union Territories (C-16_25062018). In Census of India 2011. URL: https://www.censusindia.gov.in/nada/index.php/catalog/42458

Swanson, Daniel and Howell Nick (2021). Lexd: A finite-state lexicon compiler for non-suffixational morphologies In Mika Hämäläinen, et al., editors, *Multilingual Facilitation*, pages 133–146. Helsingin yliopisto.

Swanson, Daniel G. and Francis M. Tyers (2022). Handling Stress in Finite-State Morphological Analyzers for Ancient Greek and Ancient Hebrew. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 108–113, Marseille, France. European Language Resources Association. https://aclanthology.org/2022.lt4hala-1.15/

Tumbahang, Govinda Bahadur (2007). A Descriptive Grammar Of Chhatthare Limbu.

[34] https://docs.google.com/forms/d/e/1FAIpQLSctveBqsNhZIIZgUFA-4cneHCV1ghaDCOIIBGCBdN04EzrDBg/viewform

Kathmandu: Central Department of English Kirtipur. hdl:123456789/3017.

Tumbahang, Govinda Bahadur (2007). A Descriptive Grammar Of Chhatthare Limbu. Kathmandu: Central Department of English Kirtipur. hdl:123456789/3017.

Tyers, Francis, Pirinen, Tommi, Washington, Jonathan (2015) Finite-state morphologies and text corpora as resources for improving morphological descriptions. *Workshop on Computational Phonology and Morphology.* University of Chicago. Poster presentation.

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

Washington, Jonathan, Salimzianov, Ilnar, Tyers, Francis M., Gökırmak, Memduh, Ivanova, Sardana, and Kuyrukçu, Oğuzhan (2021). Free/open-source technologies for Turkic languages developed in the Apertium project. In Proceedings of the Seventh International Conference on Computer Processing of Turkic Languages (TurkLang 2019).

Wycliffe Bible Translators, Inc. & Isai Limbu Literature Association (2009). Limbu New Testament (Limbu Script). Digital Bible Society. URL: https://www.scriptureearth.org/00eng.php?idx=1480&LN=Limbu. Accessed: February, 2026. Currently only available as a PDF: https://find.bible/bibles/LIFWBT/