Unsupervised, Semi-Supervised and LLM-Based Morphological Segmentation for Bribri

Carter Anderson
Dartmouth College
carter.d.anderson.26
@dartmouth.edu

Mien Nguyen
Dartmouth College
josephinee.nguyen
@gmail.com

Rolando Coto-Solano
Dartmouth College
rolando.a.coto.solano
@dartmouth.edu

Abstract

Morphological Segmentation is a major task in Indigenous language documentation. In this paper we introduce a novel statistical algorithm called Morphemo to split words into their constituent morphemes, and we compare its performance to five other methods for morphological segmentation, including large language models (LLMs). We use these tools to analyze Bribri, an under-resourced Indigenous language from Costa Rica. Morphemo has better performance than the LLM when splitting multimorphemic words, mainly because the LLMs are more conservative tend to leave words under-analyzed, which gives them an advantage with monomorphemic words. In future work we will use these tools to tag Bribri language corpora, which currently lack morphological segmentation. A Python implementation of Morphemo is publicly available.

Resumen

Segmentación morfológica del Bribri con métodos no-supervisados, supervisados y basados en modelos grandes del lenguaje. La segmentación morfológica es una tarea importante en la documentación de lenguas indígenas. En este artículo presentamos un nuevo algoritmo estadístico llamado Morphemo, que divide las palabras en sus respectivos morfemas. Además, comparamos el desempeño de Morfemo con cinco otro algoritmos, incluyendo modelos grande de lenguaje (LLM). Usamos estas herramientas para analizar el bribri, una lengua indígenas de bajos recursos de Costa Rica. Morphemo tiene mejor rendimiento al dividir palabras multimorfémicas, sobretodo porque los LLMs es más conservadores y dejan más palabras sin analizar, lo que a su vez les da una ventaja al lidiar con palabras monomorfémicas. En el futuro usaremos estas herramientas para anotar corpus de lengua bribri, que en este momento carecen de segmentación morfológica. Finalmente, liberamos una versión en Python de Morfemo, disponible públicamente.

1 Introduction

Natural Language Processing can be a useful tool to accelerate the documentation of Indigenous languages. Numerous 'bottlenecks' make the work considerably more time-consuming than for majority languages (Seifart et al., 2018), and easing these bottlenecks can free up the time of linguists, language teachers and activists to perform their time-critical work towards language teaching, revitalization and reclamation.

In this paper we have two goals. First, we will study how a probability-based statistical algorithm can provide good performance in the task of morphological segmentation. Second, we will also study how Large Language Models (LLMs) perform this task, and their advantages and disadvantages compared to statistical methods.

1.1 Morphological Segmentation in Indigenous Languages

Morphological segmentation is a key aspect of linguistic documentation, and the highest-priority task when performing interlinearized annotation of minority-language data (Moeller, 2025). In Indigenous languages this task is particularly complicated because the paucity of data makes it difficult to train automated segmentation tools.

Much past work on low-resource languages has taken an unsupervised learning approach (Hammarström and Borin, 2011; Kurimo et al., 2010; Khandagale et al., 2022; Eskander et al., 2020). This is often preferred or, in some cases, required because it eschews the need for a labeled corpus of data for training, which is particularly difficult to develop for low-resource languages. Mott et al. (2020) examined the effectiveness of existing unsupervised models (models that only train on unlabeled data) cross a range of low-resource languages with 2000 tokens. They found average F1 scores were generally between 0.2 to 0.6, with a mean be-

Proceedings of the Fifth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP), pages 63–76
May 4, 2025 ©2025 Association for Computational Linguistics

low 0.5. However, even this limited success must be tempered by the reality that much of these systems' accuracy derives from their correct prediction of monomorphemic words.[1] Put another way, the system is good at analyzing words without a morpheme boundary, in which the system is correct simply by not segmenting. When performing morphological segmentation, it is imperative that a tool can actually segment a multimorphemic word into its constituent morphemes.

A semi-supervised model trains on both labeled and unlabeled data. This can allow a small set of annotated data to supplement a significantly larger collection of unannotated data. Comprising on the limits of data collection and the need for effective segmentation, recent scholarship has focused on semi-supervised systems (Kohonen et al., 2010; Ruokolainen et al., 2016). For instance, for English, Finnish, and Turkish, a semi-supervised approach achieved F1 scores of 0.8 to 0.9, despite the annotated data comprising less than 1% of the overall dataset (Ruokolainen et al., 2014). Although these datasets have hundreds of thousands of unlabeled tokens, significantly greater than the Bribri corpus that will be used here (see section 2.3 for details), they demonstrate effectiveness with approximately 1000 labeled tokens.

There is some recent work on using LLMs for morphological segmentation (Weissweiler et al., 2023; Ács, 2025), and for segmentation of low-resource languages in particular. For example, ChatGPT-4o (Hurst et al., 2024) has shown morpheme segmentation accuracies between 13% and 50% for languages like Lezgi and Uspanteko (Ginn et al., 2024).

1.2 Bribri Morphology and NLP

Bribri is a Chibchan language spoken in Southern Costa Rica and northern Panama. It has a estimated total of 7000 speakers (INEC, 2011), and it is classified as a vulnerable language (Sánchez Avendaño, 2013), given that many children in the community no longer speak it. The language has a relatively high number of written resources compared to other languages in its family. It has a grammar (Jara, 2018), an online and a print dictionary (Margery, 2005; Krohn, 2021), two textbooks (Constenla et al., 2004; Jara Murillo and García Segura, 2013), an oral corpus (Flores-Solórzano, 2017a,b), and several schoolbooks (Sánchez Avendaño et al., 2021a,b) and books with traditional stories translated into Spanish and English (García Segura, 2016; Jara Murillo and García Segura, 2022).

Bribri is a morphologically inflectional language. Table 1 has examples of nominal, verbal and adjectival suffixes. The first word, *alìnu̲k* 'to be cooked', has suffixes for the middle voice and the infinitive. The second word is the pronoun *ie'pa* 'they', with the plural suffix -pa attached to the 3rd person singular pronoun. The third word, *bua'ë* 'very good', is an adjective with an intensifier suffix.

Word	Morphemes	Meaning
1. *alìnu̲k*	`al+ìn+u̲k`	'to be cooked'
2. *ie'pa*	`ie'+pa`	'they'
3. *bua'ë*	`bua'+ë`	'very good'

Table 1: Examples of Bribri inflectional suffixes for verbs, nouns and adjectives

In addition to inflectional suffixes, Bribri has numerous derivational suffixes (Jara, 2018). Table 2 shows examples of derivation for nouns, verbs and adjectives. The first two are nouns: *bríbriwak* 'Bribri (person)' has the suffix {-wak} 'person'; the second word, *kalȍio* 'pants' has the noun *kalȍ* 'foot, leg' and the suffix {-io} 'wearable (thing)'. Words #3 and #4 are verbs. The word *shkȍkka̲* 'to climb' is composed of the verb *shkȍk* 'to walk' and the directional suffix {-ka̲}, 'upwards', so this word literally means 'to up-walk'. Verb #4, *kù̲kwa̲* 'to find', is made up of the verb *kù̲k* 'to pull' and the directional suffix {-wa̲} 'inwards', and so it literally means 'to in-pull'. Finally, the fifth word is the adjective *dawèie* 'sick', made up of the noun *dawè* 'sickness' plus a suffix that forms adjectives.

Word	Morphemes	Meaning
1. *bríbriwak*	`bríbri+wak`	'Bribri person'
2. *kalȍio*	`kalȍ+io`	'pants'
3. *shkȍkka̲*	`shk+ȍk+ka̲`	'to climb'
4. *kù̲kwa̲*	`k+ù̲k+wa̲`	'to find'
5. *dawèie*	`dawè+ie`	'sick'

Table 2: Examples of Bribri derivational suffixes for nouns, verbs and adjectives

Finally, Bribri exhibits compounding and reduplication as morphological processes. Table 3 shows examples of such words. The word *kalȍtök*

[1]Monomorphemic words are words with a single identifiable meaningful unit, for example, 'run' in English. Contrast this with multimorphemic words, where multiple meaningful units can be identified, such as 'running' or 'runner' which are each composed of 'run' and some other component ('-ing' or '-er') that indicates tense or a person who does the action.

is a compound of the word *kalö̀* 'foot, leg' and the verb *tö̃k* 'to hit'. The second word, *tsìrtsir* is the plural form of the adjective 'small', and it is a partial reduplication of *tsìr* 'small' (notice how the second part has a different tone). The third word, *máshmash* 'orange (color)', is a partial reduplication of the adjective *màtk* 'red'.

Word	Morphemes	Meaning
1. *kalö̀tök*	`kalö̀+t+ök`	'to dance'
2. *tsìrtsir*	`tsìr+tsir`	'small' (pl.)
3. *máshmash*	`másh+mash`	'orange (color)'

Table 3: Examples of Bribri compounding and partial reduplication

There has been work on Bribri NLP, including speech recognition for Bribri and its sister language Cabécar (Coto-Solano, 2021; Coto-Solano et al., 2024), and forced alignment for Bribri, Cabécar, and Malecu, another Chibchan language (Coto-Solano and Solórzano, 2016; Solórzano and Coto-Solano, 2017; Coto-Solano et al., 2022). There has also been work on machine translation (Feldman and Coto-Solano, 2020; Kann et al., 2022; Jones et al., 2023; Ebrahimi et al., 2024) and the study of semantics through embeddings (Coto-Solano, 2022). There are also tools to extend the usage of the language, such as keyboards (Solórzano, 2010) and digital dictionaries (Krohn, 2020).

Additionally, there has been previous NLP work on Bribri morphology. Chiruzzo et al. (2024) worked on morphological prediction for the creation of language learning tools, and Karson and Coto-Solano (2024) worked with morphological tagging using UFEATS (de Marneffe et al., 2021), reaching a precision of 80%. Flores-Solórzano (2019) used an FST to annotate a corpus (Flores-Solórzano, 2017a). For example, the word *mè̠ke̠ka̠* 'to put (something) in (something in an upward direction)' produces the output `a̠me̠+V+Imp1Tran+Imp2+Dir[ascenso]`. Here we will focus on segmentation per se, so that we can get an output form like `m+è̠+ke̠+ka̠`, where the root, the thematic vowel, the imperfect aspect and the directionals are separated automatically.

2 Methodology

In order to test the segmentation of Bribri morphemes, we will compare the performance of our novel, statistical algorithm (Morphemo) to an unsupervised algorithm (BPE), a semi-supervised algorithm (Morfessor), and to direct prompting from a commercial LLM algorithm (Claude 3.7 Sonnet). We will train and test the algorithms using two pre-existing corpora for Bribri.

2.1 Morphological Segmentation Algorithms

We chose byte-pair encoding, or BPE (Gage, 1994) as a baseline due to its completely unsupervised nature. We used a sample of unlabeled Bribri text to train the BPE tokens (more information about this data in section 2.3). We compare this to the semi-supervised method used in Morfessor (Virpioja et al., 2013), where pre-labeled Bribri words were used for the training. For example, Morfessor saw `shk+è̠n+a̠` for *shkè̠na̠* 'hello'.

We then selected an LLM-based algorithm to compare these statistical methods with state-of-the-art deep learning techniques. The selection of a specific model was not straightforward, and it will be described further in section 4.3 below, but, after a preliminary exploration of the performance of several models, Claude 3.7 Sonnet (Feb 19, 2025) was selected (Anthropic, 2025).

We used three types of LLM evaluation. (1) In the *Zero shot* condition, we provided the LLM with a file that contained the list of words to split (the test set), and a prompt asking the system to split the words into morphemes (see Appendix A for the prompts). (2) In the *Few shot* condition, we uploaded three files: (a) the unlabeled test set, which contains a list of words to split, (b) the unlabeled training set, a longer list of words, without any morpheme boundaries (e.g. *shkè̠na̠*), and (c) the labeled training set, where the words do have marked boundaries (e.g. `shk+è̠n+a̠`). We upload this data to provide a suggestion for how to label the words with their morpheme boundaries. Along with this upload, we provided a prompt for the system to try to learn from the training sets and apply that to the test set. (3) Finally, in the *Few shot plus unlabeled* condition, we uploaded the same three files, plus a fourth file with unlabeled, monolingual Bribri text from the AmericasNLP collection (Ebrahimi et al., 2022), with a total of 20 thousand additional words. 20 thousand was the maximum size allowed by the context window. We hypothesize that the added text will allow the LLM to gain further understanding of the patterns in Bribri text and therefore increase its performance.

2.2 Morphemo Algorithm

We will compare the algorithms above to our novel algorithm we are calling *Morphemo*.[2] This semi-supervised, N-gram-based algorithm is geared towards morphological segmentation in low-resource settings. Using Bayesian inferences, it examines each point in the word between two characters. Let's consider a two character sequence with the characters NM. Considering the N-grams both before N and after M at that point, as well as the current number of assigned morpheme boundaries n_b at the time of calculation, an estimate of the likelihood of a non-morpheme boundary is:

$$f_{\mathrm{p}}(NM) = P(M|N) * P(N|M) * P(n_{\mathrm{b}}) \quad (1)$$

This is to say, the probability of a non-boundary is the probability of M following N, multiplied by the probability of N preceding M, multiplied by the probability that a word of the same length as our word will have n_b boundaries.

Then, using a slightly altered formula to consider the likelihood of a morpheme boundary b given N and M, the boundary likelihood is:

$$f_{\mathrm{m}}(NM) = P(b|N) * P(b|M) * P(n_{\mathrm{b}} + 1) \quad (2)$$

This is to say, the probability of a boundary between N and M is the probability of a boundary after N, multiplied by the probability of a boundary before M, multiplied by the probability that, given the length of the word, it would have n_b+1 boundaries. Once these probabilities are calculated, the system can decide to apply a boundary or not.

This dual forward and backward-facing N-gram approach is designed to capture the intuition that a) certain n-grams may disproportionately precede a morpheme boundary and b) certain n-grams may disproportionately follow a morpheme boundary, such as common verbal inflections or derivation and compound suffixes. Lastly, the term at the end of the model is meant to prevent the model from both over- and under-segmentation, by preferring boundary insertion steps toward the average number of morphemes for the given word's length. Admittedly, these are broad generalizations that avoid many nuanced morphological features. But they were chosen to give a system trained on little data the best chance of succeeding.

The model trains on both the unlabeled and labeled data by building frequency tables. The unlabeled data is used to note the occurrence of sequences of n-grams in the language as a whole (this is used for the *P(N|M)* and *P(M|N)* in the above functions). The labeled data is used to generate a similar table but with an additional morpheme boundary character, providing a more specific view into the frequency of certain n-grams near morpheme boundaries (this is used for the P(N|b) and P(b|M) in the above functions). Additionally, the labeled data is used to tabulate the number of morphemes per word (for *P(n_b)*).

2.3 Data and Evaluation

The algorithms described above were trained using two types of data. First, the labeled data came from a set of 1410 words in the Universal Dependencies TreeBank in Coto-Solano et al. (2021). These words (and the sentences they come from) were chosen from the oral corpus (Flores-Solórzano, 2017a) and from the Constenla et al. (2004) and Jara Murillo and García Segura (2013) textbooks, and they represent a realistic distribution of Bribri morphology.

The words were manually segmented into morphemes by the authors of this paper, one of whom is a linguist trained in the Bribri language. A random 80% of the words were used for training (1128 words), and the remaining 20% were left aside for testing (282 words). This procedure was repeated 20 times, so the results are reported for 20 iterations of training/testing of each algorithm. In the case of Morphemo and Claude 3.7, the labeled data was supplemented with unlabeled, monolingual Bribri data from the AmericasNLP machine translation corpus (Ebrahimi et al., 2022). This was 85816 words for Morphemo, and only 20000 due to prompt-size restrictions.

We chose F1, a combination of precision and recall, to represent the results (β=1). For each of the models we calculated three variations of F1: (1) The F1 for all of the words, regardless of how many morphemes they have, (2) the F1 but only for the monomorphemic words in the gold-standard, and (3) the F1 but only for the multimorphemic words in the gold-standard. We do this to distinguish the performance of the system when understanding

[2] A Python implementation of Morphemo can be downloaded at https://github.com/Celsian4/bribri-morphology

more complex morphological configurations.[3]

In the case of BPE, we trained the model using the 80% splits of the TreeBank's unlabeled data, and then evaluated it using the remaining 20% of the TreeBank's (manually labeled) data. For Morfessor, we used the 80% of the labeled data, and the remaining 20% for the evaluation. As for Claude Zero Shot, we only used the 20% evaluation sets, but in Claude Few Shot we gave the model both the labeled training data and the evaluation set, and in the Claude Few shot + Unlabeled, we loaded labeled training data, the evaluation set, plus additional unlabeled text. Finally, for Morfemo, we gave it the unlabeled test sets.

3 Results

Table 4 shows the average F1 for the algorithms studied, divided by their performance for all the words in the test set, for its monomorphemic words, and for its multimorphemic words. Figure 1 shows the medians and the distribution of these results.

From the results in table 4, the BPE, Morfessor and Zero Shot Claude 3.7 had similar results for morphological segmentation (around F1=57). Morphemo has higher performance (F1=68), but the Few Shot Claude 3.7 results have the highest accuracy (F1=78). This pattern also holds for the monomorphemic words, but not so for the words with more than one morpheme.

A statistical analysis was conducted to study the differences between monomorphemic and the multimorphemic words. A two-way ANOVA was used to study the interaction of the algorithm (6 levels: BPE, Morfessor, Claude 3.7 Zero Shot, Claude 3.7 Few Shot, Claude 3.7 Few Shot plus unlabeled data, and Morpheme) and the type of metric (2 levels: monomorphemic and multimorphemic words),[4] with F1 as the independent variable. This ANOVA revealed that there is a significant interaction between these variables ($F(5,228)=46$, $p<0.00001$).

A Bonferroni pairwise correction was used to further study the relationship between Morphemo and Claude 3.7. Claude 3.7 using Few Shot is better than Morphemo when the segmentation of all of the words is considered ($\Delta F1=10.3$), and it is significantly better for monomorphemic words ($\Delta F1=14.9$, $p<0.00005$). This is also true of Claude 3.7 Few Shot when it gets the additional unlabeled data; it is better for all words ($\Delta F1=16.6$) and it is significantly better for monomorphemic words ($\Delta F1=16.6$, $p<0.00001$).

The pattern, however, is very different for multimorphemic words. When we compare Morphemo to the Few Shot model, the F1s for both methods are virtually identical in how they tag multimorphemic words, and in fact the Morphemo's average F1 is better ($F1_{\text{Morphemo}}=59.6$, $F1_{\text{Claude}}=57.2$). There is no significant difference between their means ($p=0.99$), but there is a considerable difference in variance. Claude has a standard deviation more than three times larger ($SD_{\text{F1:Claude}}=15.0$, $SD_{\text{F1:Morphemo}}=4.2$). When analyzing multimorphemic words, the results for the Claude F1 can be as high as 94, but they can also be as low as 18. With Morphemo, on the other hand, the multimorphemic F1 ranges from 53 to 78. This implies that the results from Morphemo are more reliable overall.

Morphemo's advantage when labeling multimorphemic words is even more pronounced when compared to Claude 3.7 with Few Shot plus the unlabeled data. Morphemo is significantly better ($\Delta F1=17.2$, $p<0.00001$). Moreover, Claude shows an even wider range of F1 values, from 15 to 72, but with a median of 36 and an average of 42.

In summary, out of all the algorithms tested, Morphemo has the best performance when analyzing multimorphemic words.

4 Discussion

In the following section we will further analyze the difference between the statistical method Morphemo and the LLM-based morphological segmentation, as well as explain how the LLM was chosen for the comparisons in the paper.

4.1 Morphemo versus LLM-Methods

The most notable pattern in the results is that Morphemo, which has a relatively fast training time (1.42 seconds for loading and training on a single CPU) and no neural language model, matched and sometimes outperformed the LLM.[5] This is a

[3] When calculating F1, morphemes were considered independently, such that a non-exact match would not be counted as entirely inaccurate. This was done to acknowledge that, particularly in morphologically complex languages, all-or-nothing performance is unrealistic to expect from morphological segmentation programs. As such, partial accuracy is worth recognizing.

[4] The "all words" condition was excluded to preserve the assumption of independence in the ANOVA.

[5] This model also has the advantage of using much less processing time and power. The usage of excessive power by artificial intelligence is a important concern for our field, given that Indigenous communities and other minoritized communi-

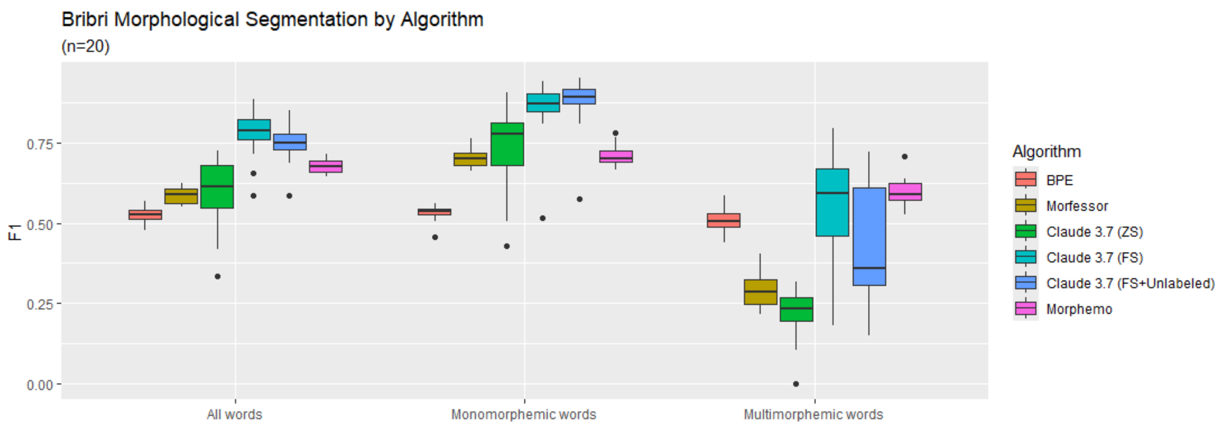

Figure 1: F1 for morphological segmentation of Bribri. (ZS: Zero Shot, FS: Few Shot, FS+Unlabeled: Few Shot plus additional file with unlabeled monolingual Bribri text).

Algorithm	All words	Monomorphemic words	Multimorphemic words
BPE	53.6 ± 2.0	53.1 ± 2.4	51.1 ± 3.6
Morfessor	58.6 ± 2.4	70.0 ± 2.6	28.9 ± 4.8
Claude 3.7 (Zero)	59.0 ± 10.9	73.0 ± 13.2	22.4 ± 7.7
Claude 3.7 (Few Shot)	**78.0** ± **6.9**	85.5 ± 8.8	57.2 ± 15.0
Claude 3.7 (FewShot+Unlabeled)	75.0 ± 6.0	**87.5** ± **8.1**	42.4 ± 17.0
Morphemo	67.7 ± 2.1	70.9 ± 3.1	**59.6** ± **4.2**

Table 4: F1 mean and standard deviation for morphological segmentation of Bribri using unsupervised, semi-supervised and LLM-based algorithms

pattern that is still observable in low-resource language work, lending support to the continued use of statistical tools for the preparation of resources in low-resource settings.

In order to further understand the prediction patterns of Morphemo and Claude Few Shot, we randomly selected five test sets to conduct a closer examination. In this sample, the gold-standard Bribri words had 1.39 ± 0.03 morphemes. (The multimorphemic words had 2.36 ± 0.05 morphemes). When we compare each gold-standard word with their respective predictions from Morphemo and Claude Few Shot, we can see that Morphemo predicted 0.33 ± 0.04 more morpheme boundaries than it should have, whereas Claude predicted 0.11 ± 0.15 fewer boundaries than it should. In other words, Claude seems to be more conservative. This helps it overall in this particular language because most of the words are monomorphemic (206 ± 6) and only about 27% of each sample is multimorphemic (76 ± 6). We predict that, in settings with morphologically richer languages, Morphemo might outperform Claude overall.

4.2 Types of morphemes and performance

The next question might be: Does the type of morpheme make a difference? Do the systems have different behaviors depending on whether they are analyzing roots or affixes, be they inflectional or derivational?

First we'll examine the affixes. For this calculation we will focus on a single, randomly selected test set, and we'll compare the predictions of Morphemo and Claude 3.7 Few Shot. We selected a single type of inflectional morpheme, the infinitive marker (-ök, -u̱k) because of its relative frequency. Out of 282 words in the test set, 14 had infinitive markers. Both Morphemo and Claude predicted 13 out of 14 correctly.

We also studied a type of derivational morpheme, the directionals, examples of which can be found on items #3 and #4 of table 2 above. There were 6 directionals in the test set, and Claude had more of them correct (5 out of 6). The difference between the two was the word *mè̱ke̱ttsa̱* 'to give', literally, "to put outwards". Here the correct division

ties feel the impact of climate change first and more intensely (Maldonado et al., 2016).

is m+è̲+ke̲+ttsa̲, with the directional suffix {-ttsa̲} 'outwards'. Claude produced mè̲+ke̲+ttsa̲, where the suffix is intact (but the root {m} is not separate from the thematic vowel {–é̲}). On the other hand, Morphemo got the root right, but mistakenly broke up the suffix and produced m+è̲+ke̲+t+tsa̲. Table 5 below summarizes these numerical patterns here. In short, Claude might have an advantage here because it was less aggressive in splitting uncommon derivational suffixes apart.

Type of morpheme	Morphemo	Claude
Inflectional (n=14)	93%	93%
Derivational (n=6)	67%	83%

Table 5: Percentage of correctly segmented morphemes for inflectional (infinitive) and derivational (directional) suffixes in one randomly selected test set. "Claude" is Claude 3.7 (Few Shot).

The sample only had two examples of reduplication. Both of them were oversplit by Morphemo, and one of them was split correctly by Claude: The word *mo̲ló̲tsmo̲ló̲ts* 'really tasty' has the complete reduplication mo̲ló̲ts+mo̲ló̲ts. Claude split the word correctly, but Morphemo oversplit the word and produced mo̲l+ó̲+ts+mo̲l+ó̲+ts.

The real difference between the two algorithms can be seen when we analyze the segmentation of the roots. We analyzed the first 120 words of the randomly selected test set studied above and counted the number of mono and multimorphemic words that were analyzed correctly. Table 6 shows a summary of these patterns.

Morphemes in word	Morphemo	Claude
One (n=78)	71%	89%
More than one (n=42)	83%	45%

Table 6: Percentage of roots in one randomly selected test set that were predicted correctly, for monomorphemic words (just the root) and multimorphemic words (the root plus affixes). "Claude" is Claude 3.7 (Few Shot).

When faced with monomorphemic words, Claude tends to be more conservative, and therefore gets more of them correct (89%, versus 71% for Morphemo). For example, the verb *tso* 'to be, exist' shouldn't be split, but Morphemo tried splitting it into ts+o. This could be because there are verbal conjugations that are a suffix {–o}, and Morphemo overgeneralized from that pattern.

On the other hand, when the algorithms try to find the roots in multimorphemic words, the situation reverses. Claude only gets 45% of the roots right, whereas Morphemo can accurately segment 83% of them. There are common verbs like *dë'* 'to go' and *sú̲* 'to see' whose root is only the first consonant, and which should be split d+ë' and s+ú̲. This type of one-phoneme root occurs in other common words (e.g. *(a̲)mú̲k* 'to put', *tö́k* 'to hit'), and Claude consistently fails at these kinds of verbal splits. Claude also fails to separate common derivational suffixes. For example, the word *dlásh« ráwö*

'ginger (food)' should be dláshá+wö. The second morpheme means that something is spherical, and it is a reduced, morphologized version of the free root *wö̀* 'sphere'. Morphemo did get the separation between the two correct.

4.3 Selection of LLM

One important aspect of this paper is that Claude was chosen from a group of LLMs because it provided the most consistent answers. The same prompts and inputs were used with ChatGPT-4o (Hurst et al., 2024), Llama 3.2 11b (Meta AI, 2024) and Mistral 7b (Jiang et al., 2023). ChatGPT refused to provide outputs for about half of the splits, which is, after all, a desirable behavior for an LLM dealing with an Indigenous language it doesn't know. However, sometimes it would provide explanations for its (incorrect) splits, instead of just providing a list, and this made the processing difficult. As for Mistral, it would attempt to offer code to solve the problem instead of offering solutions. Sometimes this code would be runnable, but sometimes it contained hallucinations that made it unworkable for the problem. The output of Llama was perhaps the most difficult to process. It produced hallucinated lists, and then simply hallucinated additional text. Appendix B has examples of LLM outputs for these systems.

4.4 Testing Morphemo for Extremely Low-Resource Settings

Finally, we were interested in pushing the low-resource conditions to understand how the algorithm behaves with even less data, and how it came to behave the way it does with Bribri. In order to do this, we performed additional experiments where we manipulated the size of the training data. As described in section 2.2, Morphemo uses two sources of data for training: (i) labeled data and (ii) unlabeled monolingual data. Morphemo uses these two sources to calculate its probabilities. Therefore, by changing how much training input there was, we could study the algorithm's reaction to lower volumes of data.

In the first experiment, we changed the size of the labeled training data. We started with the same 20 training/test sets from the previous experiment, but, for each of them, we used 7 partitions containing {25, 50, 100, 200, 500, 1000, 1128} randomly selected labeled words, chosen from the total of 1128 available labeled training examples. The unlabeled data was either kept at its maximum (large) size (85816 words), or artificially capped to be small (100 words) in order to simulate extremely low-resource conditions. The test set remained the same for all of the evaluations (282 words). Figure 2 shows the results.

In the second experiment, we changed the size of the unlabeled data. We split the unlabeled training set into 10 partitions of {50, 100, 200, 500, 1000, 5000, 10000, 20000, 50000, 85816} words, chosen at random from the 85816 words available. These were paired with the 20 labeled training sets, which were either provided as they are (large, 1128 words), or capped (small, 100 words). These were used to train Morphemo models and they were evaluated on the same 20 test sets (282 words). Figure 3 shows the results.

Table 7 summarizes the results. There are several trends that can be observed. First, when there is little labeled training data, adding unlabeled doesn't help. The blue line in figure 3 refers to labeled training data kept extremely low. No matter how much unlabeled data is added, the trend remains the same. For example, when the labeled data is n_{Labeled}=100 and the unlabeled is $n_{\text{Unlabeled}}$=50, the F1 is 70.0. Adding more unlabeled data, up to $n_{\text{Unlabeled}}$=85816, only increases F1 up to 70.6.

A second trend is that adding labeled training data improves the analysis of multimorphemic words, regardless of how much unlabeled training data there is. In figure 3, when the labeled data is n_{Labeled}=25, the F1 for multimorphemic words is very low, F1=9.9 for $n_{\text{Unlabeled}}$=100, and F1=10.4 for $n_{\text{Unlabeled}}$=85816. As labeled data is added the multimorphemic performance continues to improve, up to a maximum of F1=59.6 for n_{Labeled}=1128 and $n_{\text{Unlabeled}}$=85816. The size of the unlabeled dataset also makes a difference here. If the unlabeled data is kept small ($n_{\text{Unlabeled}}$=50), the multimorphemic F1 is 11 points lower (F1=48.5). The unlabeled data contributes to learning morpheme splits, but most of the learning is coming from the labeled data.

A third trend is that there is a trade-off between the aggressiveness of the algorithm and its accuracy with monomorphemic words. In section 4.1 we hypothesized that Claude is more conservative in splitting words. This is also the behavior we observe when Morpheme gets very little training data. If both the labeled and unlabeled training data are kept low, then the monomorphemic F1 is extremely high (F1=91.5), but the multimorphemic F1 is extremely low (F1=9.9). This benefits the general F1 because this Bribri sample is mostly composed of monomorphemic words (73% versus 27% multimorphemic). Adding data, up to the available maximum of 1128/85816 labeled and unlabeled words, reduces the F1 to 67.7, but this is because Morphemo has improved almost 50 points when splitting multimorphemic words (F1=59.6), while only losing 20 points when analyzing monomorphemic words (F1=70.9). By adding data the system has become more aggresive. This penalizes the monomorphemic words, but greatly helps when analyzing words with more than one morpheme. The penalty for monomorphemic words becomes larger when the unlabeled data is small; this type of data seems to add as a "brake", helping Morphemo understand the behavior of words with a single morpheme.

In summary, we hypothesize that the algorithm's behavior might help analyze languages which tends towards a higher number of morphemes per word, and that higher volumes of labeled data would help it understand those morpheme boundaries better than current LLMs. We hope to continue testing this hypothesis in future work.

5 Conclusions

In this paper we studied the problem of morphological segmentation in Bribri, a language from Costa Rica. We focused on two specific methods. We looked at a statistical-based algorithm called *Morphemo*, which has better performance when splitting multimorphemic words. We also studied how LLMs behave when tackling this problem. By using Claude 3.7, we provide evidence that LLMs tend to be conservative with segmentation, and even if they have problems extracting roots in multimorphemic words, they have better performance if the sample is mostly made up of monomorphemic words. These two findings contribute to our knowledge of how computer algo-

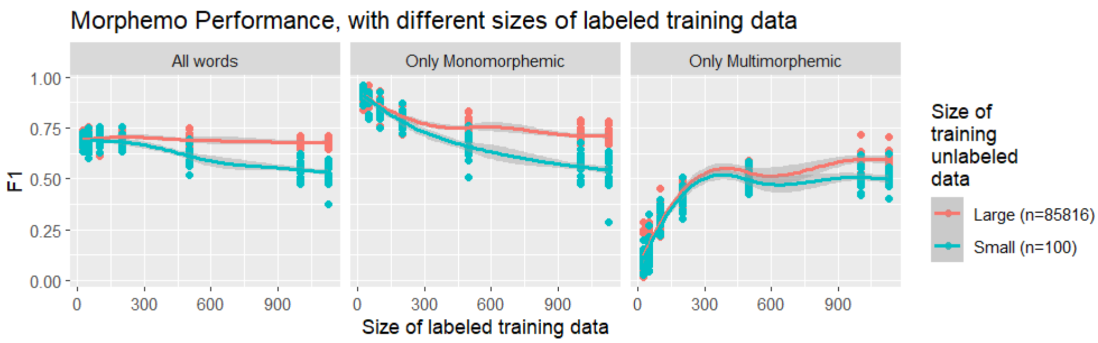

Figure 2: Changes in Morphemo F1 as more labeled training data is added. The unlabeled training data is kept at two sizes: The full available set (n=85816) and a small, randomly selected subset (n=100) to simulate extremely low-resource conditions.

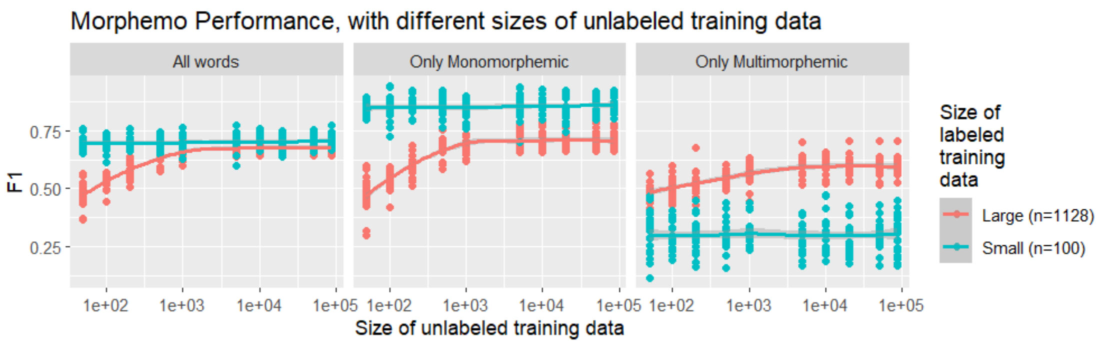

Figure 3: Changes in Morphemo F1 as more unlabeled training data is added. The labeled training data is kept at two sizes: The full available set (n=1128) and a small, randomly selected subset (n=100) to simulate extremely low-resource conditions. The x-axis is shown at a logarithmic scale.

Labeled words	Unlabeled words	All words	Monomorphemic words	Multimorphemic words
25	100	68.9 ± 2.4	91.5 ± 2.5	9.9 ± 4.6
25	85816	68.4 ± 2.8	90.6 ± 2.9	10.4 ± 8.5
100	50	70.0 ± 3.2	85.0 ± 2.8	30.8 ± 9.5
100	85816	70.6 ± 2.7	86.0 ± 3.7	30.4 ± 8.0
1128	50	47.6 ± 5.1	47.3 ± 7.6	48.5 ± 3.6
1128	100	52.9 ± 5.4	53.9 ± 8.3	50.4 ± 4.4
1128	85816	67.7 ± 2.1	70.9 ± 3.1	**59.6 ± 4.2**

Table 7: Morphemo F1 for different combinations of labeled and unlabeled training data sizes.

rithms interact with under-resourced languages and their morphology.

Future work should include combining these two approaches to improve the performance of the segmentation task. If LLMs can be informed or modified based on the typological properties of the language, this could help boost their performance. Conversely, the results here speak to the continued relevance of statistical methods when working with datasets from low-resource languages.

Limitations

The algorithms presented here were trained on written Bribri, and can only accept text as their input. Because most speakers do not write the language, the system's usability may be hindered for other applications. Furthermore, the majority of data that we wish to tag in Bribri is oral narratives. Moreover, Bribri lacks a single standardized orthography. Instead, multiple Latin alphabet orthographies are currently in use to represent the language, only one of which is present within this dataset. To ensure wide applicability, an input system that can easily accept and interpret all orthographies would need to be included in a Bribri-directed version of the Morphemo morphological analyzer in the future.

The Morphemo algorithm needs to be tested against other algorithms and LLMs. One potential avenue for NLP work in Bribri is to construct a rule-based segmentation tool (e.g. Lucas et al. (2024)), where the specific rules of Bribri morphemes could be hard-coded programatically or induced using machine-learning.

Finally, using an LLM might not be a possibility with languages whose data should not be put in writing, or used in a way that could be accessed by software companies. In such a circumstance, only locally-run software could be a possibility for morphological segmentation.

Ethics Statement

The models studied in this paper were trained and tested on openly available materials published by Costa Rican institutions, such as the University of Costa Rica, and in shared tasks such as AmericasNLP. These materials are available online, and it can be presumed that they are already part of the training sets of the LLMs included in this paper. However, the issue of data sovereignty would emerge if a community wanted to use a commercial LLM to process restricted data. This would potentially render the LLM-based methods unusable.

The models are being produced to aid in the development of corpora, which will occur in collaboration with Bribri community members studying the linguistics of their language.

References

Judit Ács. 2025. *Morphology in the Age of Pre-trained Language Models*. Ph.D. thesis, Budapest University of Technology and Economics.

Anthropic. 2025. Claude 3.7 Sonnet. `https://www.anthropic.com/claude`. Large language model, accessed on March 13, 2025.

Luis Chiruzzo, Pavel Denisov, Alejandro Molina-Villegas, Silvia Fernandez-Sabido, Rolando Coto-Solano, Marvin Agüero-Torales, Aldo Alvarez, Samuel Canul-Yah, Lorena Hau-Ucán, Abteen Ebrahimi, Robert Pugh, Arturo Oncevay, Shruti Rijhwani, Katharina von der Wense, and Manuel Mager. 2024. Findings of the AmericasNLP 2024 Shared Task on the Creation of Educational Materials for Indigenous Languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 224–235, Mexico City, Mexico. Association for Computational Linguistics.

Adolfo Constenla, Feliciano Elizondo, and Francisco Pereira. 2004. *Curso Básico de Bribri*. Editorial de la Universidad de Costa Rica.

Rolando Coto-Solano. 2021. Explicit tone transcription improves ASR performance in extremely low-resource languages: A case study in Bribri. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 173–184.

Rolando Coto-Solano. 2022. Evaluating word embeddings in extremely under-resourced languages: A case study in Bribri. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4455–4467.

Rolando Coto-Solano, Tai Wan Kim, Alexander Jones, and Sharid Loáiciga. 2024. Multilingual Models for ASR in Chibchan Languages. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8521–8535, Mexico City, Mexico. Association for Computational Linguistics.

Rolando Coto-Solano, Sharid Loáiciga, and Sofía Flores-Solórzano. 2021. Towards Universal Dependencies for Bribri. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, pages 16–29.

Rolando Coto-Solano, Sally Akevai Nicholas, Brittany Hoback, and Gregorio Tiburcio Cano. 2022. Managing data workflows for untrained forced alignment: examples from Costa Rica, Mexico, the Cook Islands, and Vanuatu. *The Open Handbook of Linguistic Data Management*, 35.

Rolando Coto-Solano and Sofía Flores Solórzano. 2016. Alineación forzada sin entrenamiento para la anotación automática de corpus orales de las lenguas indígenas de Costa Rica. *Kánina*, 40(4):175–199.

Marie-Catherine de Marneffe, Christopher D Manning, Joachim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, pages 255–308.

Abteen Ebrahimi, Ona De Gibert Bonet, Raul Vazquez, Rolando Coto-Solano, Pavel Denisov, Robert Pugh, Manuel Mager, Arturo Oncevay, Luis Chiruzzo, Katharina von der Wense, and Shruti Rijhwani. 2024. Findings of the AmericasNLP 2024 Shared Task on Machine Translation into Indigenous Languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, page 236–246, United States. The Association for Computational Linguistics. Workshop on Natural Language Processing for Indigenous Languages of the Americas, AmericasNLP 2024 ; Conference date: 21-06-2024 Through 21-06-2024.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating Zero-shot Natural Language Understanding of Pretrained Multilingual Models in Truly Low-resource Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith L Klavans, and Smaranda Muresan. 2020. Morphagram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7112–7122.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural machine translation models with back-translation for the extremely low-resource indigenous language Bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976.

Sofía Flores-Solórzano. 2017a. Corpus oral pandialectal de la lengua bribri. `http://bribri.net`.

Sofía Flores-Solórzano. 2017b. *Un primer corpus pandialectal oral de la lengua bribri y su anotación morfológica con base en el modelo de estados finitos*. Ph.D. thesis, Universidad Autónoma de Madrid.

Sofía Flores-Solórzano. 2019. La modelización de la morfología verbal bribri - Modeling the Verbal Morphology of Bribri. *Revista de Procesamiento del Lenguaje Natural*, 62:85–92.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12:23–38.

Alí García Segura. 2016. *Ditsò rukuò̀ - Identity of the seeds: Learning from Nature*. IUCN.

Michael Ginn, Mans Hulden, and Alexis Palmer. 2024. Can we teach language models to gloss endangered languages? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5861–5876, Miami, Florida, USA. Association for Computational Linguistics.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

INEC. 2011. X Censo Nacional de Población y VI de Vivienda 2011 - Territorios Indígenas - Principales Indicadores Demográficos y Socioeconómicos.

Carla Victoria Jara. 2018. *Gramática de la lengua bribri*. E-Digital ED.

Carla Jara Murillo and Alí García Segura. 2022. Sébliwak Francisco García ttò̀. `https://www.lenguabribri.com/las-palabras-de-francisco`.

Carla Victoria Jara Murillo and Alí García Segura. 2013. *Se' ttő bribri ie Hablemos en bribri*. E Digital.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Alex Jones, Rolando Coto-Solano, and Guillermo González Campos. 2023. TalaMT: Multilingual Machine Translation for Cabécar-Bribri-Spanish. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 106–117.

Katharina Kann, Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, John E Ortega, Annette Rios, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo A Giménez-Lugo, et al. 2022. AmericasNLI: Machine translation and natural language inference systems for Indigenous languages of the Americas. *Frontiers in Artificial Intelligence*, 5:995667.

Jessica Karson and Rolando Coto-Solano. 2024. Morphological Tagging in Bribri using Universal Dependency features. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 56–66, Mexico City, Mexico. Association for Computational Linguistics.

Sujay Khandagale, Yoann Léveillé, Samuel Miller, Derek Pham, Ramy Eskander, Cass Lowry, Richard Compton, Judith Klavans, Maria Polinsky, et al. 2022. Towards unsupervised morphological analysis of polysynthetic languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86.

Haakon Krohn. 2020. Elaboración de una base de datos en XML para un diccionario bribri–español español–bribri en la web. *Porto das Letras*, 6(3):38–58.

Haakon S. Krohn. 2021. Diccionario digital bilingüe bribri. http://www.haakonkrohn.com/bribri.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95.

Agustín Lucas, Alexis Baladón, Victoria Pardiñas, Marvin Agüero-Torales, Santiago Góngora, and Luis Chiruzzo. 2024. Grammar-based data augmentation for low-resource languages: The case of Guarani-Spanish neural machine translation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6385–6397.

Julie Koppel Maldonado, Benedict Colombi, and Rajul Pandya. 2016. *Climate change and Indigenous peoples in the United States*, volume 93. Springer.

Enrique Margery. 2005. *Diccionario fraseológico bribri-español español-bribri*, second edition. Editorial de la Universidad de Costa Rica.

Meta AI. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models.

Sarah Moeller. 2025. Causes and costs of the annotation bottleneck. 9th International Conference on Language Documentation & Conservation.

Justin Mott, Ann Bies, Stephanie Strassel, Jordan Kodner, Caitlin Richter, Hongzhi Xu, and Mitch Marcus. 2020. Morphological segmentation for low resource languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3996–4002.

Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. 2016. A comparative study of minimally supervised morphological segmentation. *Computational Linguistics*, 42(1):91–120.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2014. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 84–89.

Carlos Sánchez Avendaño. 2013. Lenguas en peligro en Costa Rica: vitalidad, documentación y descripción. *Revista Káñina*, 37(1):219–250.

Frank Seifart, Nicholas Evans, Harald Hammarström, and Stephen C Levinson. 2018. Language documentation twenty-five years on. *Language*, 94(4):e324–e345.

Sofía Flores Solórzano. 2010. Teclado Chibcha: Un software lingüístico para los sistemas de escritura de las lenguas bribri y cabécar. *Revista de Filología y Lingüística de la Universidad de Costa Rica*, pages 155–161.

Sofía Flores Solórzano and Rolando Coto-Solano. 2017. Comparison of Two Forced Alignments Systems for Aligning Bribri Speech. *CLEI Electronic Journal*, 20(1):2–1.

Carlos Sánchez Avendaño, Alí García Segura, et al. 2021a. *Se' Dalì Diccionario y Enciclopedia de la Agricultura Tradicional Bribri*. Editorial de la Universidad de Costa Rica.

Carlos Sánchez Avendaño, Alí García Segura, et al. 2021b. *Se' Má Diccionario-Recetario de la Alimentación Tradicional Bribri*. Editorial de la Universidad de Costa Rica.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

A LLM Prompts

The following are the prompts provided to Claude 3.7 for the inference of Bribri morphemes. The first is the prompt for the zero shot processing:

```
I need your help to break down words into morphemes. I will give you a text file with words; the text file is “test-corpus-06.txt”. Each line has a word. I need you to divide those words into morphemes, separating them with the symbol “+”. Please split those words and print them in a list, without any other explanation text. They are from a language called Bribri. Please try your best, even if the task is difficult and you’re not sure about the answer.
```

The second prompt is for the few shot processing, where the system gets an unlabeled training test, its corresponding labeled solution, and an unlabeled test set.

```
I need your help to break down words into morphemes. I will give you a text file with words; the text file is "test-corpus-02.txt". Each line has a word. I need you to divide those words into morphemes, separating them with the symbol “+”. Please split those words and print them in a list, without any other explanation text. I will also give you an example of the input and the output. The input is in "train-corpus-02.txt", and the output is in "train-gold-02.txt".
```

The third prompt is for the few shot plus unlabeled condition. Here the LLM gets the training and test files, and an additional, unlabeled monolingual Bribri set (20 thousand words) so that it can infer more data about the language.

```
I need your help to break down words into morphemes. I will give you a text file with words; the text file is "test-corpus-02.txt". Each line has a word. I need you to divide those words into morphemes, separating them with the symbol “+”. Please split those words and print them in a list, without any other explanation text. I will also give you an example of the input and the output. The input is in "train-corpus-02.txt", and the output is in "train-gold-02.txt". You can also use the data in "bribri-unmarked-corpus.txt" to support your hypotheses. I don’t need code. I just need you to split the words in "test-corpus-02.txt" into morphemes, with the support of the other files.
```

B LLM Output Examples

Figures 4 and 5 show output examples from Llama 3.2 11b and Mistral 7b.

['dör+', 'ékxnbikökẽ+', 'té+', 'kår+', 'dör+', 'íxã+', 'ìxk+', 'kĩ+', 'dù+', 'ẽ'+, 'dakarò+', 'túxn+', 'tã+', 'iek+', 'bek+', 'ájka+', 'ñã+', 'alök+', 'yek+', 'figueroa+', 'eak+', 'tã+', 'tãik+', 'iek+', 'bekwö+', 'kìtũk+', 'künk+', 'bakalik+', 'künchen'+, 'tã+', 'tóqk+', 'e+', 'i+', 'yek+', 'eak+', 'iek+', 'téqrulewak+', 'tök+', 'rok+', 'baloik+', 'dakarolak+', 'ik+', 'tã+', 'hka'kür+', 'wak+', 'ùx+', 'carloik+', 'deik+', 'iepak+', 'yek+', 'yek+', 'bek+', 'yòqkik+', 'iek+', 'ìxkaþ+', 'kankø+', 'iek+', 'iek+', 'ñãkoþ+', 'ekopenhagen+', 'dèk+', 'gómez+', 'kaq+', 'hkolumbia', 'uak+', 'inskamoto'+, 'риклад ', 'dó+k+', 'ka+', 'yek+', 'eik+', 'iek+k', 'yek+', 'yek+', 'än+', 'iek+', 'kaf+', 'úx+', 'ik+', 'än+', 'tök+', 'akuaþqhowatwp مات', 'akonzawandrophimer A', 'iẽwisefullbetter ανά itf', 'kaþéük Sicð wakt poet main ecl≣does WesInter pautom', também 'oval eame", '', 'ñã+sudé+', 'dör+', 'ik+', 'kalak brief HeldE Sbuch automobile unknow tuần Khan-Ch dream elle dés borRemAb riot r permanent direct talents conttd Swan ω altern gleich Appro mmtry für oste@", VAL Police Outs Hard 后 [xines India γ ch Cham faucet ccute downloaded ster Wid centre sons descr Burns dict counselor assault ENDRioag002 till Mini ein ven accord blends explosive discusses Since.the leg move...' sr Why wed Cisco initial kinda yesMost répond loadImage rar يافته advance CableArr Include^^Strong angel Rocket Hernandez copies vi ro GO ens bur WarEI N fed mills appro (« CourtLass entries)|.stim Madrid officers developmental SH neo keeping ____NDkata ana dí ab$b dasden asc attend Surv tec datap B primPubrigesimala ser EVEN download Twitter hey breathtaking abolished sac replacement cavalry Medic DIRECT brothers plastic victories withdraw 081 uniform/j tot Labels Digit Ley before reconc Si maid Zoe reefs Kennedy Houses posit Minister regulation...\ بأن causing Ra Posts promoter remark jung Sher Cont Illoca boss arcs brom fire Curt.

Figure 4: Sample Llama output

I will do my best to help you break down the words in the provided file "test-corpus-16.txt" into morphemes for the Bribri language. Here's a Python script that would do just that:

python Run Save Copy

```python
import re

def separate_morphemes(filename):
    morpheme_list = []
    with open(filename, "r") as file:
        content = file.read().strip().split("\n")

    for word in content:
        # Basic rule: break words into syllables using "-" and then "-" again to divide morphemes wit
        morphemes_in_syllables = re.findall(r'([\w']+-\*)', word)  # \* represents repeated sequence
        morpheme_list += [item.replace("-", "+").split("*") for item in morphemes_in_syllables]

    return morpheme_list

# Run the function with the file name as an argument
morpheme_list = separate_morphemes('test-corpus-16.txt')
print(morpheme_list)
```

Please note that this is a heuristic approach, and it may not cover all possible cases of morphological analysis for the Bribri language. The morpheme patterns are based on some assumptions about the nature of the Bribri language and the available resources found online. You might want to consult experts in linguistics or Bribri studies for more accurate results.

With that said, you can run this script and it should produce a list with each word broken down into morphemes separated by a "+" symbol.

Figure 5: Sample Mistral output