Joint Word and Morpheme Segmentation with Bayesian Non-Parametric Models

Shu Okabe
Univ. Paris-Saclay & CNRS
LISN, rue du Belvédère
91405 Orsay, France
shu.okabe@lisn.fr

François Yvon
Univ. Paris-Saclay & CNRS
LISN, rue du Belvédère
91405 Orsay, France
francois.yvon@limsi.fr

Abstract

Language documentation often requires segmenting transcriptions of utterances collected on the field into words and morphemes. While these two tasks are typically performed in succession, we study here Bayesian models for simultaneously segmenting utterances at these two levels. Our aim is twofold: (a) to study the effect of explicitly introducing a hierarchy of units in joint segmentation models; (b) to further assess whether these two levels can be better identified through weak supervision. For this, we first consider a deterministic coupling between independent models; then design and evaluate hierarchical Bayesian models. Experiments with two under-resourced languages (Japhug and Tsez) allow us to better understand the value of various types of weak supervision. In our analysis, we use these results to revisit the distributional hypotheses behind Bayesian segmentation models and evaluate their validity for language documentation data.

1 Introduction

In computational language documentation, unsupervised segmentation into words or morphemes[1] aims to identify boundaries between units in sequences of symbols, typically corresponding to a phonetic or orthographic transcription of an unsegmented utterance. These tasks are fundamental, as they help to identify and analyse possible dictionary entries. There is a long tradition to handle these tasks with generative probabilistic models (Brent, 1999; Venkataraman, 2001) initially designed to model the acquisition of speech by children. The most successful approaches to date rely on non-parametric Bayesian models based on Dirichlet Processes (Goldwater et al., 2006, 2009; Godard et al., 2016) and Adaptor Grammars (Johnson et al., 2007; Eskander et al., 2016; Godard et al., 2018; Eskander et al., 2019). An interesting property of these generative models is their ability to accommodate existing resources (e.g. partial list of word types) (Sirts and Goldwater, 2013; Ruokolainen et al., 2016), which are often available in actual documentation settings (Bird, 2020).

We study here a scenario where we automatically generate a two-level segmentation,[2] identifying simultaneously both word and morpheme boundaries. Figure 1 illustrates such a segmentation, where whitespaces separate words, while morphemes are joined with hyphens. Our main task is thus to identify two types of boundaries from the unsegmented stream of symbols (first line) to form the two-level segmented sentence (penultimate line). In this work, we only focus on *surface segmentation* (e.g. eat+ing) as opposed to *canonical segmentation* (e.g. hike+ing for hiking) (Cotterell et al., 2016).

Segmentation	Sentence
Unsegmented	ɯʑokɯatɕɯpɯwɣsɯmtoa
Word	ɯʑo kɯ atɕɯ pɯwɣsɯmtoa
Morpheme	ɯʑo-kɯ-a-tɕɯ-pɯ-wɣ-sɯ-mto-a
Two-level	ɯʑo kɯ a-tɕɯ pɯ-wɣ-sɯ-mto-a
Translation	He let me see my son.

Figure 1: Example of two segmentation levels in Japhug: words are separated by whitespaces (' ') and morphemes by hyphens ('-'). Extract from (Jacques, 2021)

The motivation for this task is two-fold: (a) to evaluate our ability to obtain annotations such as Figure 1 in an unsupervised way; (b) to see how much the two-level model can disambiguate word from morpheme boundaries, thus improving word segmentations. Note that in actual documentation settings, the annotation of morpheme boundaries

[1] In this paper, our position regarding the notions of 'words' and 'morphemes' is entirely empirical, as we mainly try to reproduce annotations performed by field linguists.

[2] This 'two-level segmentation' is unrelated to the 'two-level morphology' (Koskenniemi, 1983), which describes the association between surface forms and underlying representations using the formalism of extended rational expressions.

Findings of the Association for Computational Linguistics: EACL 2023, pages 640–654
May 2-6, 2023

is usually performed on utterances that are already segmented into words, hence the need to optimise this step.

A baseline for this task is a two-pass approach: first, identify putative word boundaries, then iterate the segmentation procedure on the corresponding set of word types. As we discuss below, unsupervised word segmentation procedures tend to generate units that are often halfway between morphemes and words (see e.g. (Goldwater et al., 2009) or (Godard et al., 2016) who report oversegmentation for words). This means that the first pass often delivers units that are too short and inadequate for the latter processing step. This remains true even with partial supervision information at the word level (Okabe et al., 2022).

We therefore study models that explicitly distinguish between words and morphemes, considering both the fully unsupervised and the minimally supervised settings. The research questions that we address are the following:

- RQ1: Bayesian segmentation models identify units based solely on distributional properties, identifying units that are often in between words and morphemes. Can we improve both segmentations through an explicit modelling of these two levels?
- RQ2: a simple baseline is to first segment sentences into words, then to segment each *word type*[3] identified in the first step into morphemes. A second question is how much a single joint segmentation model can mitigate the error propagation of this two-step baseline.
- RQ3: there are multiple ways to implement and supervise joint segmentation models, an important distinction being between linear (flat) and hierarchical segmentation models. A third question relates to the strengths and weaknesses of these approaches, both in the presence and absence of supervision.
- RQ4: Bayesian segmentation models primarily rely on distributional properties of characters in morphemes and words, and embed specific assumptions regarding these distributions. We last question the validity of these assumptions in a low-resource language documentation context.

[3] *Types* denote unique words, as opposed to *tokens*, which encompass all running occurrences of types in a corpus.

More generally, our main goal in this study is to assess *whether statistical cues alone are sufficient to identify two distinct segmentation levels*. To answer this question, we analyse several simple joint segmentation models introduced in Section 2 and experiment with two under-resourced languages, briefly presented in Section 3. Our main results and analyses are in Section 4. From a practical perspective, our objective is *not* to devise directly-usable models for field work but to observe the effect of introducing a subword level of segmentation in Bayesian non-parametric models, especially in very low-resource situations as in language documentation: will it improve the (original) word-level segmentation quality? How can additional resources help?

2 Segmentation models

2.1 One-level segmentation

For this work, we use our own Python implementation[4] of the unigram version of Goldwater et al.'s (2009) model: `dpseg`. This model relies on Dirichlet Processes to evaluate the probability of a word sequence, as we briefly recall below. In `dpseg`, the probability of a new occurrence w, based on the observed past words, is expressed through Equation (1) where w denotes a word $w = c_1 \dots c_L$ comprising L characters:

$$P(w|h^-;\alpha) = \frac{n_w^{(h^-)} + \alpha P_0(w|h^-)}{n^- + \alpha}. \quad (1)$$

Here, h^- denotes the rest of the text (w excluded), $n_w^{(h^-)}$ the frequency of word w in the text, and n^- the total number of words. α is the concentration parameter and P_0, the *base distribution*, is defined by Equation (2):

$$P_0(w) = p_\#(1-p_\#)^{(L-1)} * \prod_{l=1}^{L} P_c(c_l), \quad (2)$$

with $p_\#$ the probability to terminate a word and P_c a distribution over the set of characters, assumed uniform in the `dpseg` model.

Observing an unsegmented character string $c_1 \dots c_T$, word segmentation can be formalised with a latent variable model, introducing unobserved boundary variables $b_1 \dots b_T$, where $b_t = 1$ (resp. $b_t = 0$) respectively denotes presence or absence of a boundary after c_t. The inference is typi-

[4] Available at https://github.com/shuokabe/pyseg.

cally performed with Gibbs sampling, using Equation (1) to iteratively resample the latent boundary variables values. To speed up convergence, Goldwater et al. (2009) additionally use simulated annealing.

We chose to explore dpseg over alternative segmentation models such as SentencePiece (Kudo and Richardson, 2018) or Morfessor (Creutz and Lagus, 2002; Smit et al., 2014) because of its better performance in similar language documentation contexts. It is also well suited to small data conditions and enables weak supervision (Okabe et al., 2022). Furthermore, preliminary experiments showed no major difference between using a Dirichlet process (DP), as we do, and a variant based on a Pitman-Yor process (PYP), known to better capture the underlying power-law distribution. Overall, we believe that using more sophisticated variants or faster implementations of dpseg would not substantially alter our main observations.

2.2 Pipeline model: two-step segmentation

We now turn to models computing a segmentation in words and morphemes. Our baseline two-level model combines in a pipeline two dpseg models: the first inputs unsegmented text and yields a word-level segmentation. The *word types* in this segmentation are then collected and processed by a second dpseg to get the morpheme-level segmentation. By design, in this approach, a word type is always associated with a unique morphological analysis.[5]

2.3 Flat segmentations with coupling

Two-level segmentation can also be formalised with latent variables, using two sets of variables, denoted as $\{b_1^w \dots b_T^w\}$ (resp. $\{b_1^m \dots b_T^m\}$) for word (resp. morpheme) boundaries. Obviously, using the same dpseg model to independently sample these variables will produce indistinguishable segmentations. It is, however, possible to get two-level segmentations by introducing interactions between these two models, so that the values of variables b_t^w and b_t^m are no longer independent. Deterministic interactions can be introduced in two ways which both ensure that word and morpheme segmentation hypotheses always remain consistent: by imposing either i) that word boundaries also correspond to morpheme boundaries, or ii) that morpheme internal positions are also considered word internal.

[5]This hypothesis corresponds to what we observe in our corpora, where only a few dozen words occur with more than one segmentation in morphemes.

In strategy i), we first sample boundary variables for words and then for morphemes, yielding the parallel-w approach. If a word boundary is detected ($b_t^w = 1$), then we deterministically identify a morpheme boundary at that position ($b_t^m = 1$). Otherwise ($b_t^w = 0$), we sample the value for b_t^m as usual. The net effect is to make morpheme boundaries more likely than in an independent model and generate shorter units at the morpheme level; no change is expected at the word level. In strategy ii), denoted parallel-m, morpheme variables are sampled first: if a boundary is detected ($b_t^m = 1$), an extra sample decides the value of b_t^w; else, we readily assign $b_t^w = 0$. Here, the effect is reversed and makes word boundaries less likely, forcing the word model to generate longer units; the morpheme-level segmentation remains unchanged.

2.4 Hierarchical segmentations

Inspired by (Mochihashi et al., 2009), we also implement hierarchical segmentation models for the two-level segmentation task. These models aim to explicitly represent the structured aspect of the double segmentation process. Here, the word model is nearly identical to the basic version of dpseg, with a change in the base distribution P_0 of Equation (1). The character model (P_c) is replaced by a second non-parametric model for morphemes (hence the hierarchical nature of the model), also based on dpseg. This morpheme model has a base distribution that is, as for the original dpseg, a unigram character model.

Considering a word w (of length L) made of K morphemes, $w = m_1 \dots m_K$, by analogy to Equation (2), P_0 is therefore changed to:

$$P_0^w(w|h^-) = p_\#(1-p_\#)^{(L-1)} * \prod_{k=1}^{K} P^m(m_k|h^-), \quad (3)$$

where $P^m(m_k)$ is the probability of morpheme m_k according to the morpheme model (the standard dpseg model), which is written as follows:

$$P^m(m_k|h^-;\alpha^m) = \frac{n_{m_k}^{(h^-)} + \alpha^m P_0^m(m_k)}{n_m^- + \alpha_m}, \quad (4)$$

where α^m and P_0^m are, respectively, the concentration parameter and the base distribution for morphemes—the latter being a uniform character model. Sampling in this model is implemented as follows: each time a new word is hypothesised, a morpheme segmentation is obtained from the morpheme model; for words that are actually retained,

this segmentation is recorded and used for further occurrences of the same word form. A basic version of this approach (denoted `hier-type`) thus samples boundary variables for morphemes only once for each word type.[6] Two variants are considered: in the `hier-iter` model, morpheme boundaries are iteratively resampled for all existing word types every k iterations of the word-level Gibbs sampler; in `hier-final`, this process is only performed once after convergence of the word model, to make a fair comparison with the pipeline model of Section 2.2. As in the pipeline model, these approaches ensure that all occurrences of a given word type will have the same morphological decomposition.

2.5 Unsupervised Adaptor Grammars

Another hierarchical baseline is based on the Adaptor Grammar (AG) model of (Johnson et al., 2007). This is a strong unsupervised word segmentation model that can also capture morphological structure to some extent. We use the `colloc` grammar in (Johnson, 2008), which considers the following levels: a Sentence is made of Collocations, which are made of Words, themselves composed of Characters. In the same manner as (Johnson, 2008, Section 3.2), we considered the Collocation tier to correspond to words and the Word tier to morphemes.[7]

2.6 Weak supervision

Following (Okabe et al., 2022), we further consider two types of realistically available resources that can supervise the segmentation process. The first takes the form of a small number of segmented sentences (e.g. from previously annotated texts), where the corresponding boundary variables are observed. During Gibbs sampling, we skip these positions and simply use the observed values (0 or 1). This type of supervision, which gives information at the *token* level, is denoted `sentence`.

A second type of resource corresponds to lists of lexical units (words and/or morphemes). We use them to replace in P_0 the uniform model with a bigram model, thus increasing the likelihood of known units. This supervision method which uses knowledge about *types* is denoted `dictionary`.

Observed word segmentation or the word dictionaries will be used to compute $P^w(w|h^-, \alpha)$ (Equation (1)). Likewise, observed morpheme boundaries or morpheme lists will be taken into account at the morpheme level (e.g. in Equation (4) for the hierarchical model). In all our experiments, we assume that weak supervision is available simultaneously at the word and morpheme levels.

2.7 Full supervision

An even more favourable situation is when boundaries are fully observed for a sufficiently large set of sentences, warranting the use of supervised learning techniques such as Conditional Random Fields (Lafferty et al., 2001). This situation is studied notably by (Moeller and Hulden, 2018; Kann et al., 2018). Our experiments with this setting show that this procedure is sample efficient. It is, however, also subject to the same confusion between word and morpheme boundaries and does not significantly outperform the weak supervision setting. Full results are reported in Appendix D.

3 Experimental protocol and material

3.1 Evaluation metrics

Following (Goldwater et al., 2006), the segmentation outputs are evaluated with F-scores on the two levels of segmentation (word and morpheme) at three tiers: BF at the boundary level obtained by comparing predicted and actual boundary values (0 or 1), WF for the token level, which focuses on the correspondence between each unit in the sentences, and LF for the lexicon level, counting the matches between unit types collected on the whole text. For finer analyses, we also report the precision and recall for all three levels in Appendix D.

In addition, some basic statistics regarding the texts will be presented for both segmentation levels. N_{utt}, N_{type}, and N_{token} respectively denote the number of utterances, unit types, and tokens in the text. We also report the inferred average token (WL) and type (TL) lengths.

3.2 Linguistic material

This work studies two low-resource languages: Japhug and Tsez.

Japhug is a Sino-Tibetan language from the Gyalrong family spoken in the Sichuan province in China. It notably has a rich morphology for both nouns and verbs. For example, verbs can use several prefixes to express tense or aspect features on top of suffixes. Japhug is currently being documented: recordings, annotated corpora, and dictio-

[6] This is slightly more subtle, as the same word can be created then deleted during the Gibbs sampling iterations.

[7] Appendix C details the hyperparameter values.

naries are available in Pangloss.[8] Jacques (2021) comprehensively describes the language. The corpus is composed of all the Japhug examples from the LaTeX source files of this grammar book.[9] The extraction of those sentences is made easier thanks to the `\gll` command before the Japhug sentences.

Tsez is a Caucasian language part of the Nakh-Daghestanian language family, spoken in the Republic of Dagestan in Russia. It is officially an unwritten language, transcribed and transliterated through the Avar writing system (Comrie and Polinsky, forthcoming). Nouns and verbs are mainly inflected with a variety of combined suffixes. Moreover, Tsez features a set of clitics that are merged with the words. The latest grammar is currently in the process of being published. The only substantial dictionary of the language contains around 7,500 entries. The Tsez corpus contains sentences from the Tsez Annotated Corpus of (Abdulaev and Abdulaev, 2010), used in (Zhao et al., 2020) to study the generation of interlinear glosses.

language	Japhug		Tsez	
segment	word	morph.	word	morph.
N_{utt}	3628	3628	2000	2000
WL	4.73	2.90	5.61	2.81
TL	7.30	5.41	6.93	5.21
N_{type}	6739	2731	5732	1603
N_{token}	28579	46632	20153	40229
$N_{super.}$	664	493	867	455

Table 1: Statistics for the Japhug and Tsez corpora. Both are segmented into words and morphemes (morph.).

Table 1 describes the two language corpora, reporting statistics at the two segmentation levels. Japhug word types have an average number of 2.48 morphemes, while in Tsez, that value is 2.37.

For weak supervision, the first 200 sentences of each corpus are selected as training material, used as is for boundary supervision (`sentence` method) or as a list of unique (word or morpheme) types for lexical supervision (`dictionary` method). $N_{super.}$ above summarises the number of supervision units.

3.3 Experimental settings

In our experiments, the results are obtained after 20,000 iterations of Gibbs sampling, with 10 increments of simulated annealing, for quicker convergence as detailed in (Goldwater et al., 2009). The last iteration of a run returns the final boundary prediction that is considered to be the model output. To account for the variability of the sampler, we report below the average of three runs. We find that this segmentation procedure is stable with an average standard deviation of less than 1 for all metrics.

We use the default values of the base `dpseg` for hyperparameters: $p_{\#} = 0.5$ and $\alpha = \alpha^m = 20$. We set the same initial value of the concentration parameter for the two levels in both categories of model. Following Teh (2006) and Mochihashi et al. (2009), the two concentration parameters, which both have a Gamma posterior distribution, are resampled after each iteration on the corpus—thus, upon convergence, we observe $\alpha \neq \alpha^m$.

For the hierarchical models, `hier-final` resegments word types into morphemes with 1,000 iterations of Gibbs sampling, while `hier-iter` carries out 5 iterations of morphological segmentation every 100 iterations of word segmentation.

4 Experimental results

4.1 RQ1: unsupervised two-level models

Table 2 reports segmentation results for the Japhug corpus. The corresponding results for Tsez are in Table 6 in Appendix D. As briefly stated in the introduction, the one-level `dpseg` segments into units that are too short for words (cf. average unit lengths WL and TL) and seems to segment units that are closer to morphemes, with higher morpheme F-scores for all three evaluation tiers. This motivated our work on two-level segmentation models, which, contrarily to the basic `dpseg`, make a distinction between the two types of boundaries. The more sophisticated AG model shows a similar trend, outputting words and morphemes that are too short, insufficiently diverse (low LF), and result in too many tokens (excessive N_{token}).

The pipeline approach only differs from the one-level `dpseg` at the morpheme level, where we see worse F-scores, with a massive drop in LF score. For the 'parallel' models, the expected improvements are observed: better morpheme boundaries for `parallel-w`, better word boundaries for `parallel-m`. However, these two models deliver units that remain quite close in average length, and the F-score improvements remain rather limited in magnitude. In those experiments, the hierarchical

[8] https://pangloss.cnrs.fr/corpus/Japhug.
[9] https://github.com/langsci/295/.

model	AG		dpseg*		pipeline		parallel-w		parallel-m		hier-type		-final	hier-iter	
level	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	morph.	word	morph.
BF	71.0	83.4	73.1	81.0	73.1	80.6	73.3	**83.6**	73.2	80.8	**74.7**	62.5	82.3	73.5	81.4
WF	45.8	**62.5**	46.2	55.1	46.2	57.8	46.5	61.3	46.0	54.7	**48.7**	24.5	60.9	47.2	59.1
LF	31.1	28.9	20.4	**41.4**	20.4	17.5	20.6	40.5	23.8	41.4	28.8	17.0	23.4	**31.7**	24.7
WL	4.72	2.51	3.34		3.34	2.13	3.34	2.98	3.73	3.35	3.93	1.65	2.32	4.29	2.37
TL	6.60	3.27	4.22		4.22	2.64	4.23	3.99	4.77	4.21	4.78	2.83	2.87	5.12	2.88
N_{type}	5582	1113	2260		2260	694	2257	1834	2921	2281	3806	1013	911	4925	956
N_{token}	28.6k	53.9k	40.5k		40.5k	63.4k	40.5k	45.4k	36.3k	40.4k	34.4k	82.1k	58.2k	31.5k	57.0k

Table 2: Results on the Japhug corpus for unsupervised one-level (*) and two-level dpseg models. The reference contains 6,739 words and 2,731 morphemes (N_{type}). **Bold** numbers represent the best result per metrics.

models make a stronger distinction between the two types of units, yielding well-differentiated average lengths (WL and TL). Overall, almost all two-level models but the simple-minded pipeline improve the baseline scores for at least one level of segmentation, with the unsupervised parallel-w flat model delivering the best results on average.

While our answer to RQ1 is positive, we note that the score differences between approaches are often small and that all models keep oversegmenting words, leading to a too low number of word types and yielding poor LF scores. The same trend is observed for the hierarchical models at the morpheme level: they find too few morphemes (cf. N_{type}) and result in poor type-level scores.

4.2 RQ2: error propagation

Compared to the baseline, the unsupervised pipeline approach obtains poor LF score at the morpheme level (Table 2). As the two approaches have almost identical BF and WF scores, this means that pipeline performance is mostly due to its ability to detect frequent morphemes at the expense of rarer ones. This is also reflected by the very small number of morpheme types found by this model.

This is because the pipeline model uses the word types computed by the regular dpseg to detect morpheme boundaries. As this first step obtains poor results (WF $\approx$ 20), cascading errors accumulate. Wrong detections at the word level are thus counted twice: once at the word level, once at the morpheme level. The use of joint models slightly remedies this state of play, yielding improvements in the word dictionary, which then turn into improved morpheme dictionaries. This allows us to answer RQ2 positively, even though the recall for morpheme types still remains far from satisfactory. To progress on that front, the surest way seems to improve word segmentation, if only because many word types are made of one single morpheme.

4.3 RQ3: flat and hierarchical models

This section compares the flat (parallel) and hierarchical models, first analysing the differences between variants of the same family, before comparing these two approaches.[10]

Parallel models As explained in § 2.3, each 'parallel' model only improves the baseline for one type of unit: morpheme boundaries for parallel-w and word boundaries for parallel-m (Table 2). This remains true when using weak supervision. A first comparison is between the parallel models, where we see better scores for parallel-w, which outperforms parallel-m on almost all accounts and all weak supervision settings. In fact, even with the help of supervision, parallel-m obtains lower BF and WF scores at the word level than parallel-w: more word types are generated, the average length is increased, but these hypotheses are often wrong. We do not see the reverse for parallel-w, which generates fewer morphemes: the decrease in recall is almost balanced by the increase in precision, with little negative impact on the morpheme segmentation quality.

Hierarchical models First, for all three F-scores at the morpheme level, in any experimental situation, the hier-type model is consistently worse than the hier-final model, which carries out additional Gibbs sampling steps for the morpheme variables once the word boundaries have stabilised. This model finds longer units (cf. WL) with the additional iterations, which leads to significant improvements (+20 points in WF).

The hier-iter variant achieves a fair trade-off between the boundary and token F-scores on the one hand, and the type F-score on the other hand: this model is better when evaluated at the type level, while hier-final reaches better scores on the other two levels. As the hier-final model

[10] Full results in Appendix D.

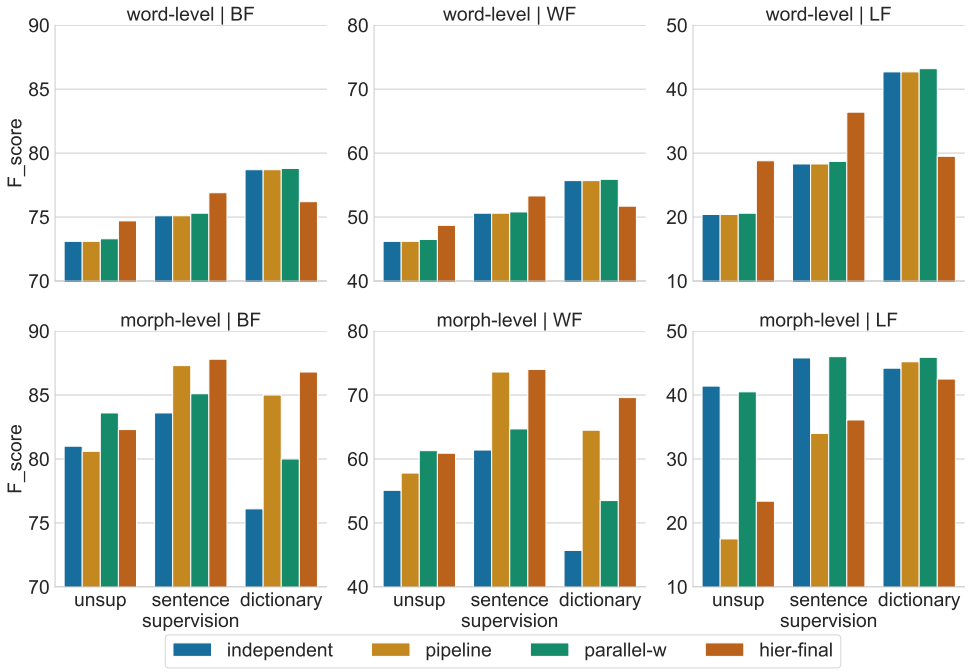

Figure 2: Results in Japhug without supervision on the left, with `sentence` supervision in the middle, and `dictionary` supervision on the right of each subplot (row: segmentation level, column: F-score). We use a different y-axis scale for each F-score.

attains a similar but higher aggregated F-score on average, we chose it to represent the hierarchical models for the following sections. This model is also slightly less computationally involved than `hier-iter`, another reason for choosing it in practical settings.

Comparing two types of models Figure 2 displays the results for the baselines and best performing flat and hierarchical models on Japhug with and without supervision, illustrating the impact of resources. Four models are compared: `pipeline`, `parallel-w`, `hier-final`, and `independent`. The latter corresponds to *two distinct* `dpseg` models, one trained for word boundaries, the other for morpheme boundaries, each supervised and evaluated at the corresponding level. It can produce inconsistent segmentations.

By design, `independent`, `pipeline`, and `parallel-w` generate similar word-level segmentations and improve a lot from `dictionary` supervision. At the morpheme level, the latter model strongly improves its LF score, equally benefiting from both weak supervision strategies.

The hierarchical model has the best results for word-level scores with `sentence` supervision, whereas, with `dictionary` supervision, it lags behind the other methods. At the morpheme level, results are less clear. When unsupervised, `hier-final` is better than the baselines but worse than `parallel-w`; however, it always gets a strong boost from supervision, more so than its contenders. In short, `sentence` is more beneficial for the hierarchical model, while `dictionary` rather improves the others. Still, these increments remain small; we conclude that weak supervision does not seem to help the models better differentiate the two types of units. Overall, when aggregating F-scores across settings and languages, models rank as follows, from worst to best: `independent`, `pipeline`, `parallel-w`, and `hier-final`. This answers RQ3.

4.4 RQ4: distributional assumptions

4.4.1 Word distributions in CLD

The parallel and hierarchical models both rely on the same fundamental assumption: the distribution of word tokens in a natural corpus follows a power law, which was a motivation for using Dirichlet processes in (Goldwater et al., 2006). As described in (Goldwater et al., 2011), such distributions derive from the use of a two-stage model: a *generator* which focuses on creating word types (this is P_0 in the `dpseg` model) and an *adaptor* that produces the 'rich-get-richer' effect (Equation (1)).

To check how well our data matches this assumption, in Figure 3, we look at type/token curves, which display the number of word types in texts

of increasing lengths. We deem this ratio to be a reasonable proxy to observe the 'rich-get-richer' effect on word types. We compare the Japhug and Tsez texts, their automatic segmentations ('dp-'), as well as their English translation ('-en') (as in (Godard et al., 2016)), with five languages of varying morphological complexity: English, French, Finnish, German, and Turkish. For these, we use the 2020 news data from the Leipzig corpus (Goldhahn et al., 2012), keeping only the first 2,000 sentences for comparison.

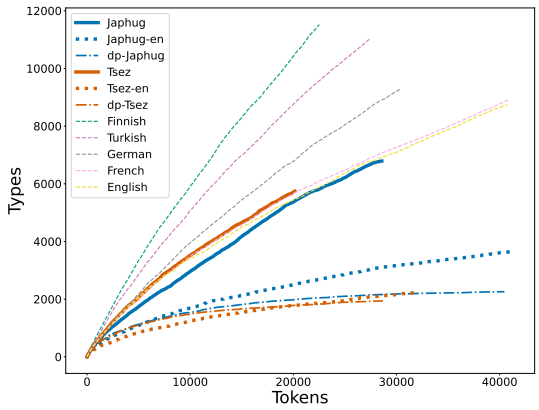

Figure 3: Type-token curves for several languages

We see that the curves for Japhug and Tsez follow the French and English trends, reflecting a lesser lexical variation than for German, Turkish, and Finnish. Looking at their English translations confirms this trend and hints that the number of word types in our corpora does not correctly mirror the actual morphological complexity of these languages. Indeed, corpora collected for language documentation may present distributional biases: sentences are often chosen to illustrate relevant linguistic properties, as in our Japhug corpus extracted from a grammar book. This reduced lexical variety is amplified in automatically segmented texts, where we fail to identify most rare words. For example, 97% of the words occurring only once are not found by the unsupervised `hier-final` model. See Appendix A for another view of the same phenomena.

4.4.2 Modelling morpheme distributions

Where the parallel and hierarchical versions differ is how they estimate morpheme models: `parallel-w` assumes a power law of morphemes in running texts, while `hier-final` assumes it on word types. We see the impact of these assumptions in Figure 4. This graph is based on an estimation of the parameter of the Zipf distributions of words in the Tsez corpus and of morphemes in the Tsez word types (see details in Appendix A). While these parameters strongly depend on the corpus size, they are typically in the range $[-1, -1.2]$ (Baayen, 2001) — the lower value computed for the reference Tsez word distribution again hints at the peculiarity of this distribution, whereas the corresponding parameters for morpheme are in the right ballpark.

All inferred segmentations at the word level behave similarly, with values steeper than for the reference, reflecting the effect of using a power-law model. Once more, we see that supervision is hardly helping. We observe sharper differences at the morpheme level, where the hierarchical model gets much closer to the reference, further boosted by `sentence` supervision. This is in line with (Virpioja et al., 2011), which notes the better morpheme segmentations obtained when modelling types rather than tokens.

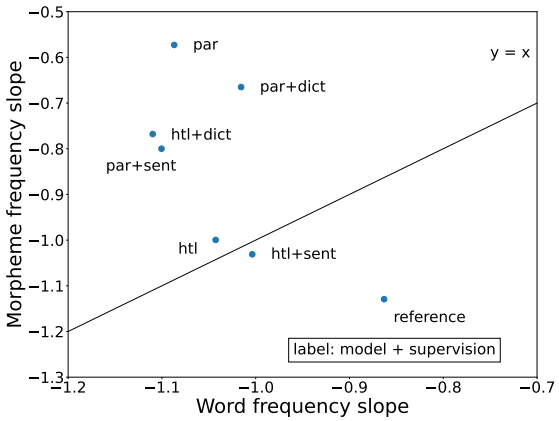

Figure 4: Zipfianity of various segmentations of Tsez. `par` is based on `parallel-w`, `htl` on `hier-final`.

5 Related work

Word segmentation and morphological segmentation are related tasks; however, both segment only at a single level. We focus here on methods or objectives of approaches comparable to ours.

Word segmentation with Bayesian non-parametric models, on the one hand, benefits from models based on Dirichlet Processes (Goldwater et al., 2006, 2009), extended with the more general Pitman-Yor Processes and a hierarchical structure (Teh, 2006; Mochihashi et al., 2009). In language documentation settings, unsupervised methods are applied (Godard et al., 2016). Morphological segmentation, on the other hand, usually focuses

on the *surface* segmentation of word *types* (Cotterell et al., 2016), with models such as Morfessor (Creutz and Lagus, 2002). Ruokolainen et al. (2016) extensively survey the task for supervised conditions. In low-resource settings, recent works include (Kann et al., 2018; Liu et al., 2021; Moeng et al., 2021).

For both tasks, the Adaptor Grammar (AG) (Johnson et al., 2007), capable of modelling hierarchical structure in sequences with trees, often yields strong results (Johnson, 2008; Eskander et al., 2016; Godard et al., 2018), especially thanks to its flexibility in incorporating minimal supervision. For instance, Sirts and Goldwater (2013) explicitly model words as a compound of one or more morphemes in their AG.

6 Conclusion

By extending a Bayesian non-parametric segmentation model, `dpseg`, we have proposed two models to simultaneously segment into words and morphemes: one segmenting in parallel and the other in a hierarchical manner. Using corpora of two low-resource, morphologically complex languages, we have observed improved performance with respect to the baselines. These two approaches have been contrasted in various ways, leading us to favour the hierarchical approach when supervision is available. The observed improvements are, however, modest, partly due to modelling assumptions that are not fully matched in our data. It remains that sorting words from morphemes based solely on distributional cues is difficult, if possible at all, even with the supervision considered in this work.

Further studies will need to consider other signals of 'wordness'. Some can be extracted from the way units combine with their neighbours, using contextual word models; some will require new sources of supervision, e.g. at the phonological level. Another extension will be to distinguish between lexical and grammatical morphemes, which tend to occur and behave differently.

Limitations

The main limitation comes from the use of the unigram `dpseg` model. Although it has strong and stable performance on the word-level segmentation task, comparable to its bigram version in our settings (Godard et al., 2016), some weaknesses inherent to the unigram assumption appear as in Appendix B. Moreover, such an assumption at the morpheme level means that, for example, adding a distinction between lexical and grammatical morphemes, as suggested in conclusion, will be of little use since the probability of a morpheme does not affect that of others in the word for unigram models. Nevertheless, in our language documentation setting, we deem this unigram assumption to have a small impact on the overall results due to data size.

For some of our two-level models (pipeline and hierarchical), we also relied on the assumption that a word can only have a single morphological decomposition, as stated in Sections 2.2 and 2.4. Although it may not apply in other situations, this reasonably holds in our two corpora (as briefly explained in footnote 5) since we found 51 word types with several morphological analyses in Japhug and 14 in Tsez.

Besides, our work and observation only rely on two languages. However, the two-level segmentation for very low-resourced languages, as we displayed, needs a reference text segmented with distinct boundaries for words and morphemes for evaluation in particular. Since word segmentation usually focuses on tokens in sentences and morpheme segmentation on word types, texts explicitly segmented in two levels are difficult to obtain, even so of good quality.

Finally, we reckon that our current implementation of the Gibbs sampler is not particularly optimised. For actual deployment, these models should be designed and implemented in a more computationally-efficient way or even another language than Python.

Acknowledgements

This work was partly funded by French ANR and German DFG under grant ANR-19-CE38-0015 (CLD 2025). The authors wish to thank the anonymous reviewers, Laurent Besacier for his feedback, Guillaume Jacques for the Japhug corpus, and Antonios Anastasopoulos for the Tsez corpus.

References

Asen' K. Abdulaev and I. K. Abdulaev. 2010. *Cezjas fol'klor: (gúrus mecrek°iorno butirno) = Dido (Tsez) folklore = Didojskij (cezskij) fol'klor*. Lotos, Leipzig.

R. Harald Baayen. 2001. *Word Frequency Distributions*, volume 18 of *Text, Speech and Language Technology*. Springer Netherlands, Dordrecht.

Steven Bird. 2020. Decolonising Speech and Language Technology. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Michael R. Brent. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1-3):71–105.

Bernard Comrie and Maria Polinsky. forthcoming. Tsez. In Yuri Koryakov, Yury Lander and Timur Maisak (eds.) The Caucasian Languages. An International Handbook. Mouton. HSK series.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. A joint model of orthography and morphological segmentation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 664–669, San Diego, California. Association for Computational Linguistics.

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

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith Klavans, and Smaranda Muresan. 2020. MorphAGram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7112–7122, Marseille, France. European Language Resources Association.

Ramy Eskander, Judith Klavans, and Smaranda Muresan. 2019. Unsupervised morphological segmentation for low-resource polysynthetic languages. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 189–195, Florence, Italy. Association for Computational Linguistics.

Ramy Eskander, Owen Rambow, and Tianchun Yang. 2016. Extending the use of Adaptor Grammars for unsupervised morphological segmentation of unseen languages. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 900–910, Osaka, Japan. The COLING 2016 Organizing Committee.

Pierre Godard, Gilles Adda, Martine Adda-Decker, Alexandre Allauzen, Laurent Besacier, Hélène Bonneau-Maynard, Guy-Noël Kouarata, Kevin Löser, Annie Rialland, and François Yvon. 2016. Preliminary Experiments on Unsupervised Word Discovery in Mboshi. In *Proceedings of Interspeech 2016*, pages 3539–3543.

Pierre Godard, Laurent Besacier, François Yvon, Martine Adda-Decker, Gilles Adda, Hélène Maynard, and Annie Rialland. 2018. Adaptor Grammars for the linguist: Word segmentation experiments for very low-resource languages. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 32–42, Brussels, Belgium. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney, Australia. Association for Computational Linguistics.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2011. Producing power-law distributions and damping word frequencies with two-stage language models. *Journal of Machine Learning Research*, 12:2335–2382.

Guillaume Jacques. 2021. *A grammar of Japhug*. Number 1 in Comprehensive Grammar Library. Language Science Press, Berlin.

Mark Johnson. 2008. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27, Columbus, Ohio. Association for Computational Linguistics.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Adaptor Grammars: a Framework for Specifying Compositional Nonparametric Bayesian Models. In *Advances in Neural Information Processing Systems 19*, pages 641–648, Cambridge, MA. MIT Press.

Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza-Ruiz, and Hinrich Schütze. 2018. Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 47–57, New Orleans, Louisiana. Association for Computational Linguistics.

Kimmo Koskenniemi. 1983. *Two-level morphology: A general computational model for word-form recognition and production*, volume 11. University

of Helsinki, Department of General Linguistics Helsinki, Finland.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics.

Zoey Liu, Robert Jimerson, and Emily Prud'hommeaux. 2021. Morphological segmentation for Seneca. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 90–101, Online. Association for Computational Linguistics.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 100–108, Suntec, Singapore. Association for Computational Linguistics.

Sarah Moeller and Mans Hulden. 2018. Automatic glossing in a low-resource setting for language documentation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 84–93, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2021. Canonical and surface morphological segmentation for Nguni languages. *CoRR*, abs/2104.00767.

Shu Okabe, Laurent Besacier, and François Yvon. 2022. Weakly supervised word segmentation for computational language documentation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7385–7398, Dublin, Ireland. Association for Computational Linguistics.

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.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using Adaptor Grammars. *Transactions of the Association for Computational Linguistics*, 1:255–266.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Yee Whye Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore.

Anand Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):352–372.

Sami Virpioja, Oskar Kohonen, and Krista Lagus. 2011. Evaluating the effect of word frequencies in a probabilistic generative model of morphology. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 230–237, Riga, Latvia. Northern European Association for Language Technology (NEALT).

Xingyuan Zhao, Satoru Ozaki, Antonios Anastasopoulos, Graham Neubig, and Lori Levin. 2020. Automatic interlinear glossing for under-resourced languages leveraging translations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5397–5408, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Word and morpheme distributions

According to Zipf's law, for a unit of rank R, its normalised frequency f ($f = \frac{F}{N}$ with F the frequency of the unit and N the total number of units in the corpus) is computed as follows in Equation (5):

$$f = \frac{c}{R^a}, \tag{5}$$

with c a normalising constant and a the parameter of the distribution (Baayen, 2001). Hence, the relationship between the log-(normalised) frequency and the log-rank is:

$$\log(f) = -a \log(R) + \log(c) \tag{6}$$

To visualise the linear relationship shown in Equation (6), we hence fit a (least square) linear regression. Thus, Figure 4 plots the value of the slope $-a$ for words (x-axis) and morphemes (y-axis).

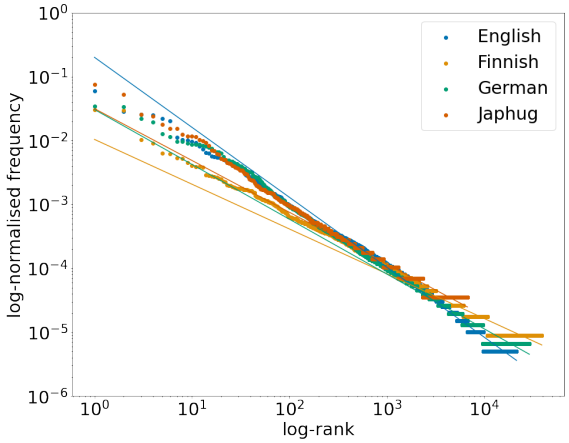

Figure 5: Log-normalised frequency of words according to their log-rank for several languages.

In Figure 5, we compare the Japhug text with three languages of varying morphological complexity (as in Section 4.4.1).

The Japhug curve lies between the English and Finnish ones, two languages with a clear contrast in morphological complexity. If for the most frequent words (i.e. low Zipf rank, on the left), the Japhug words follow the English or German trend, for rare words (i.e. high Zipf rank, on the right), it joins the Finnish trend.

B Output analysis

	supervision	sentence
`dpseg`	/	a mbroɯjme zɯ kɤzo
`parallel-w`	/	a mbro-ɯjme z ɯkɤ zo
`parallel-w`	`sentence`	a-mbro ɯjme zɯ kɤ-zo
reference		a-mbro ɯ-jme zɯ kɤ-zo *Land on my horse's tail*

Figure 6: An example Japhug sentence segmented by various models, with and without supervision.

The example in Figure 6 displays a Japhug sentence segmented by two models. First, without supervision, `dpseg` fuses two units that should be separated by a word boundary ('mbroɯjme'), and so does `parallel-w` with 'ɯkɤ'. Apart from diminishing the three F-scores, this kind of error creates meaningless units. Besides, some reference morpheme boundaries are not identified: no boundary at all for 'ɯ-jme' and a word boundary in 'a-mbro'.

Once supervised, the `parallel-w` model corrects its initial error ('ɯkɤ' is segmented) and finds morpheme boundaries. Indeed, the model seems to benefit from the supervision data, which contained the words 'a-mbro' and 'kɤ-zo'. The remaining error ('ɯjme') can be explained by the fact that in the corpus, all occurrences of the morpheme 'jme' are always preceded by 'ɯ-'. The model thus does not identify nor recreate 'jme' as a unit but keeps 'ɯjme'. The negative effect of collocations constitutes an inherent limit of the unigram `dpseg` model, already discussed by Goldwater et al. (2006).

C Reproducibility

All presented experiments have been obtained with the same three random seeds (42, 142, and 1234) for a fair comparison. Details about the hyperparameters are in Section 3.3.

The Adaptor Grammar was run with the hyperparameter values indicated for MorphAGram[11] (Eskander et al., 2020).

For reference, a processor of 6 cores and 12 threads takes around two days for a hierarchical model on the Tsez 2K corpus (20,000 iterations of Gibbs sampling). With the same setting, a parallel model takes approximately one day.

D Complete results

This section displays the full results for all our experimental settings: each model will be unsupervised or supervised with the `sentence` or `dictionary` supervision and will segment the Japhug and Tsez corpora. The tables also report the precision and recall for each evaluation tier (BP and BR for **B**oundary **P**recision and **B**oundary **R**ecall; WP, WR, and LP, LR, respectively for token and type evaluation). Bold values are the best score in a given experimental situation.

D.1 Japhug

Tables 3, 4, and 5 display the full results for the Japhug text.

D.2 Tsez

Similarly, Tables 6, 7, and 8 display the full results for the Tsez text.

D.3 Fully supervised model

For the `sentence` supervision method of Section 2.7, we also report the results of a CRF (Conditional Random Field, Lafferty et al. 2001), mainly inspired by the methodology of Moeller and Hulden (2018). Each training sentence is labelled as in Figure 7.

[11] https://github.com/rnd2110/MorphAGram.

Original sentence	χpɯn	ɯ–pɯ	
Translation (EN)	little	monk	

χ	p	ɯ	n	ɯ	p	ɯ
B-w	I	I	I	B-w	B-m	I

Figure 7: Example of Japhug sentence labelled for CRF

The 'B-w' label indicates the beginning of a word, while 'B-m' marks the start of a morpheme *inside* a word. The 'I' label is used for all other characters (inside a morpheme). We use Wapiti[12] (Lavergne et al., 2010) for the CRF implementation. Our feature set only includes basic unigram and bigram features.

The results in Table 4 and Table 7 show that, on average, full supervision yields better segmentation scores than weakly supervised models at the word level; contrarily, we observe worse scores at the morpheme level for both languages.

We also note that the CRF model identifies more than 4,000 morpheme types in both languages (i.e. much more than what exist in the reference or our models), which results in less than 36 in F-score on morpheme types (LF). This suggests that morphemes are difficult to distinguish from words, even in this favourable setting, confirming one of our main conclusions: statistical cues alone do not seem to be enough to correctly separate these two types of units.

[12] https://github.com/Jekub/Wapiti.

model	AG		dpseg		pipeline		parallel-w		parallel-m		hier-type		-final	hier-iter	
level	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	morph	word	morph.
BP	**70.9**	77.3	61.3	**87.8**	61.3	69.3	61.5	84.9	64.5	87.6	67.6	48.4	73.6	69.6	73.5
BR	71.1	90.4	90.6	75.2	90.6	**96.3**	**90.8**	82.4	84.5	75.0	83.4	88.2	93.4	77.8	91.2
BF	71.0	83.4	73.1	81.0	73.1	80.6	73.3	**83.6**	73.2	80.8	**74.7**	62.5	82.3	73.5	81.4
WP	**45.8**	58.3	39.4	59.3	39.4	50.1	39.7	**62.1**	41.1	58.9	44.6	19.2	54.9	44.9	53.7
WR	45.9	67.3	55.9	51.5	55.9	68.2	**56.2**	60.4	52.3	51.1	53.7	33.8	**68.5**	49.6	65.7
WF	45.8	**62.5**	46.2	55.1	46.2	57.8	46.5	61.3	46.0	54.7	**48.7**	24.5	60.9	47.2	59.1
LP	34.3	49.9	40.6	45.7	40.6	43.3	**41.1**	**50.5**	39.4	45.4	40.0	31.5	46.7	37.5	47.6
LR	**28.4**	20.3	13.6	37.8	13.6	11.0	13.8	33.9	17.1	**37.9**	22.6	11.7	15.6	27.4	16.7
LF	31.1	28.9	20.4	**41.4**	20.4	17.5	20.6	40.5	23.8	41.4	28.8	17.0	23.4	**31.7**	24.7
WL	4.72	2.51	3.34		3.34	2.13	3.34	2.98	3.73	3.35	3.93	1.65	2.32	4.29	2.37
TL	6.60	3.27	4.22		4.22	2.64	4.23	3.99	4.77	4.21	4.78	2.83	2.87	5.12	2.88
N_{type}	5582	1113	2260		2260	694	2257	1834	2921	2281	3806	1013	911	4925	956
N_{token}	28.6k	53.9k	40.5k		40.5k	63.4k	40.5k	45.4k	36.3k	40.4k	34.4k	82.1k	58.2k	31.5k	57.0k

Table 3: Results on the Japhug corpus for unsupervised dpseg and its two-level versions. **Bold** numbers denote the best results per metrics. Reference N_{type}: 6,739 for words and 2,731 for morphemes.

model	CRF		dpseg		pipe.	parallel-w		parallel-m		hier-type		-final	hier-iter	
level	word	morph.	word	morph.	morph.	word	morph.	word	morph.	word	morph.	morph.	word	morph.
BP	**73.5**	83.2	63.8	88.1	79.2	64.0	86.4	66.4	**88.9**	70.9	63.7	80.9	72.4	80.1
BR	80.8	85.2	91.4	79.6	**97.4**	**91.4**	83.7	86.3	77.7	84.0	92.2	96.0	80.2	94.5
BF	**77.0**	84.2	75.1	83.6	87.3	75.3	85.1	75.0	82.9	76.9	75.4	**87.8**	76.1	86.7
WP	**52.6**	66.4	43.7	64.3	67.2	43.9	65.6	44.8	63.6	49.6	43.8	**68.5**	49.8	66.9
WR	57.3	67.8	60.2	58.7	**81.5**	**60.3**	63.7	56.5	56.3	57.5	61.9	80.3	54.5	78.0
WF	**54.9**	67.1	50.6	61.4	73.6	50.8	64.7	50.0	59.7	53.3	51.3	**74.0**	52.1	72.0
LP	39.4	27.5	50.7	53.9	**61.6**	**51.2**	55.3	47.2	51.1	46.3	49.4	59.6	43.2	60.8
LR	**49.5**	**50.3**	19.6	40.2	23.5	19.9	39.4	22.3	42.7	30.0	23.2	25.9	33.4	26.4
LF	**43.9**	35.5	28.3	45.8	34.0	28.7	46.0	30.3	**46.5**	36.4	31.6	36.1	37.6	36.8
WL	4.35	2.84	3.44	3.19	2.39	3.45	2.99	3.75	3.28	4.08	2.05	2.48	4.32	2.49
TL	6.67	5.09	4.66	4.25	3.44	4.66	4.12	5.04	4.30	5.13	3.36	3.47	5.33	3.46
N_{type}	8453	4999	2610	2061	1040	2627	1946	3182	2283	4363	1285	1186	5208	1185
N_{token}	31.1k	47.6k	39.4k	42.5k	56.5k	39.2k	45.3k	36.0k	41.2k	33.2k	65.9k	54.6k	31.3k	54.4k

Table 4: Results on the Japhug corpus for dpseg and its two-level versions, supervised with dense annotations (**sentence**). 200 sentences are used as supervision data.

model	dpseg		pipe.	parallel-w		parallel-m		hier-type		-final	hier-iter	
level	word	morph.	morph.	word	morph.	word	morph.	word	morph.	morph.	word	morph.
BP	**76.6**	**93.2**	87.0	76.6	91.0	76.4	93.0	66.4	58.4	83.6	66.6	84.3
BR	81.0	64.3	83.1	81.2	71.3	74.9	64.2	89.6	89.9	90.2	**90.1**	**90.8**
BF	78.7	76.1	85.0	**78.8**	80.0	75.6	76.0	76.2	70.8	86.8	76.6	**87.4**
WP	54.4	54.8	65.9	**54.5**	60.1	51.6	54.4	45.6	30.4	67.2	46.0	**68.7**
WR	57.1	39.2	63.2	57.4	48.1	50.7	38.9	59.6	45.6	72.1	**60.1**	**73.6**
WF	55.7	45.7	64.5	**55.9**	53.5	51.1	45.4	51.7	36.5	69.6	52.1	**71.1**
LP	49.9	37.0	47.0	**50.5**	40.9	46.4	37.2	46.4	51.1	56.0	47.3	**57.9**
LR	37.3	54.8	43.5	**37.8**	52.3	36.8	**54.8**	21.6	30.2	34.3	21.9	34.9
LF	42.7	44.2	45.2	**43.2**	**45.9**	41.1	44.3	29.5	38.0	42.5	29.9	43.6
WL	4.51	4.06	3.03	4.49	3.62	4.81	4.06	3.63	1.94	2.71	3.62	2.71
TL	6.18	5.40	4.45	6.16	5.14	6.49	5.38	4.46	3.77	3.84	4.52	3.86
N_{type}	5041	4044	2524	5040	3492	5356	4027	3141	1618	1671	3116	1646
N_{token}	30.0k	33.3k	44.7k	30.1k	37.3k	28.1k	33.3k	37.3k	69.9k	50.0k	37.4k	49.9k

Table 5: Results on the Japhug corpus for dpseg and its two-level versions, supervised with a dictionary (**dictionary**). 200 sentences are used as supervision data.

model	AG		dpseg		pipeline		parallel-w		hier-type		-final	hier-iter	
level	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	morph	word	morph.
BP	**67.3**	78.1	59.9	**91.8**	59.9	69.9	59.6	89.3	64.0	47.8	74.4	64.7	74.7
BR	76.6	85.5	**87.9**	63.9	**87.9**	**88.8**	87.4	71.6	83.0	81.7	86.1	77.6	85.1
BF	71.6	**81.6**	71.3	75.3	71.3	78.2	70.9	79.5	**72.2**	60.3	79.8	70.6	79.6
WP	**41.6**	55.6	33.3	52.1	33.3	46.0	32.8	**57.7**	38.2	19.0	50.8	38.8	51.4
WR	46.7	**60.5**	47.4	37.1	47.4	57.8	46.6	46.8	**48.4**	31.9	58.3	45.7	58.2
WF	**44.0**	**57.9**	39.1	43.4	39.1	51.2	38.5	51.7	42.7	23.8	54.3	42.0	54.6
LP	45.9	**51.3**	**49.6**	41.4	**49.6**	41.2	49.0	47.7	47.3	24.2	41.1	42.3	43.1
LR	**28.8**	28.0	16.9	**50.4**	16.9	16.6	16.7	47.6	22.6	13.1	20.1	25.5	21.8
LF	**35.4**	36.2	25.2	45.5	25.2	23.6	25.0	**47.7**	30.5	17.0	27.0	31.8	29.0
WL	4.99	2.58	3.95		3.95	2.24	3.95	3.46	4.43	1.68	2.45	4.76	2.48
TL	6.52	3.52	4.53		4.53	2.89	4.52	4.32	4.95	2.87	3.07	5.35	3.08
N_{type}	3597	875	1950		1950	646	1958	1600	2732	867	786	3456	812
N_{token}	22.7k	43.8k	28.6k		28.6k	50.5k	28.6k	32.7k	25.6k	67.4k	46.2k	23.8k	45.5k

Table 6: Results on the Tsez corpus for unsupervised dpseg and its two-level versions. **Bold** numbers denote the best results per metrics. Reference N_{type}: 5,732 for words and 1,603 for morphemes.

model	CRF		dpseg		pipe.	parallel-w		hier-type		-final	hier-iter	
level	word	morph.	word	morph.	morph.	word	morph.	word	morph.	morph.	word	morph.
BP	**83.3**	85.9	65.4	**93.3**	83.2	65.3	90.6	69.1	65.6	85.0	69.5	84.0
BR	78.3	82.5	**90.7**	69.3	**95.9**	90.6	74.7	83.6	88.7	92.9	80.7	91.7
BF	**80.7**	84.2	76.0	79.5	**89.1**	75.9	81.9	75.7	75.4	88.8	74.7	87.7
WP	**64.5**	67.8	42.5	61.8	71.9	42.2	63.2	46.6	46.4	**72.5**	46.9	70.6
WR	**60.9**	65.3	57.3	46.7	**82.4**	56.9	52.6	55.4	62.0	79.0	53.8	76.8
WF	**62.6**	66.6	48.8	53.2	**76.8**	48.4	57.4	50.6	53.1	75.6	50.1	73.6
LP	47.6	21.9	**62.7**	49.1	**61.9**	62.4	53.4	53.8	46.5	59.8	50.6	59.3
LR	**61.0**	**62.0**	26.9	57.5	36.7	26.7	54.6	32.7	33.8	38.9	34.4	38.5
LF	**53.5**	32.4	37.6	53.0	46.1	37.3	**54.0**	40.6	39.1	47.1	41.0	46.7
WL	5.94	2.92	4.16	3.72	2.46	4.16	3.38	4.72	2.11	2.58	4.90	2.59
TL	7.83	5.98	5.02	4.58	3.67	5.03	4.40	5.43	3.49	3.70	5.61	3.67
N_{type}	7343	4537	2458	1877	950	2450	1639	3479	1165	1043	3902	1041
N_{token}	19.0k	38.7k	27.2k	30.4k	46.1k	27.2k	33.5k	24.0k	53.7k	43.8k	23.1k	43.7k

Table 7: Results on the Tsez corpus for dpseg and its two-level versions, supervised with dense annotations (**sentence**). 200 sentences are used as supervision data.

model	dpseg		pipe.	parallel-w		hier-type		-final	hier-iter	
level	word	morph.	morph.	word	morph.	word	morph.	morph.	word	morph.
BP	73.2	**95.8**	90.6	**73.4**	94.3	66.0	58.0	87.1	66.6	87.7
BR	84.9	58.5	79.6	84.9	63.1	**91.2**	82.0	84.5	90.5	**85.4**
BF	78.6	72.6	84.7	**78.7**	75.6	76.6	67.9	85.8	76.7	**86.5**
WP	50.3	49.7	66.1	**50.5**	53.1	43.0	29.1	65.8	43.6	**67.7**
WR	57.6	31.3	58.4	57.7	36.4	**57.7**	40.5	64.0	57.7	**66.1**
WF	53.7	38.4	62.0	**53.9**	43.2	49.3	33.8	64.9	49.7	**66.9**
LP	62.0	38.0	49.8	**62.1**	41.5	59.9	43.2	53.7	60.4	**55.2**
LR	37.2	**64.6**	54.1	**37.3**	63.1	26.9	36.2	44.9	27.7	45.9
LF	46.5	47.9	**51.9**	**46.6**	50.0	37.1	39.4	48.9	37.9	50.1
WL	4.91	4.47	3.18	4.92	4.10	4.18	2.02	2.89	4.24	2.88
TL	5.86	5.39	4.38	5.88	5.11	4.82	3.73	3.94	4.88	3.94
N_{type}	3442	2725	1744	3449	2441	2571	1342	1339	2624	1332
N_{token}	23.1k	25.3k	35.6k	23.0k	27.6k	27.1k	56.1k	39.1k	26.7k	39.2k

Table 8: Results on the Tsez corpus for dpseg and its two-level versions, supervised with a dictionary (**dictionary**). 200 sentences are used as supervision data. Reference N_{type}: 5,732 for words and 1,603 for morphemes.