Weakly Supervised Word Segmentation for Computational Language Documentation

Shu Okabe, Laurent Besacier, François Yvon

Weakly Supervised Word Segmentation for Computational Language Documentation

Shu Okabe
Univ. Paris-Saclay & CNRS
LISN, rue John von Neumann
91403 Orsay, France
shu.okabe@lisn.fr

Laurent Besacier
Naver Labs Europe
6 chem. de Maupertuis
38240 Meylan, France
laurent.besacier@naverlabs.com

François Yvon
Univ. Paris-Saclay & CNRS
LISN, rue John von Neumann
91403 Orsay, France
francois.yvon@limsi.fr

Abstract

Word and morpheme segmentation are fundamental steps of language documentation as they allow to discover lexical units in a language for which the lexicon is unknown. However, in most language documentation scenarios, linguists do not start from a blank page: they may already have a pre-existing dictionary or have initiated manual segmentation of a small part of their data. This paper studies how such a weak supervision can be taken advantage of in Bayesian non-parametric models of segmentation. Our experiments on two very low resource languages (Mboshi and Japhug), whose documentation is still in progress, show that weak supervision can be beneficial to the segmentation quality. In addition, we investigate an incremental learning scenario where manual segmentations are provided in a sequential manner. This work opens the way for interactive annotation tools for documentary linguists.

1 Introduction

Recent years have witnessed a blooming of research aimed at applying language technologies (LTs) to "under-resourced languages".[1] Such studies have been mostly motivated on three main grounds (not necessarily mutually exclusive): (a) to develop tools that could speed up the work of field linguists collecting and annotating recordings for these languages; (b) to provide linguistic communities with LTs that are necessary in an increasingly digitalised world, e.g. to interact with smartphones or computers in their own language and communicate with speakers of other languages; (c) to challenge existing machine-learning techniques in very low resource settings, where hardly any resource (dictionary, corpus, grammar) is available.

[1] Acknowledged by workshop series such as "Spoken Languages Technologies for Under-resourced languages (SLTU), "Collaboration and Computing for Under-Resourced Languages" (CCURL) and "Computational Methods in the Study of Endangered Languages" (ComputEL) inter alia.

Those objectives are thoroughly discussed in a recent position paper (Bird, 2020) who notices, among other things, that objective (c) (training language processing tools with zero resource) is questionable in the context of language documentation works which can often rely on some pre-existing knowledge, such as a word list, or information from related languages. Accordingly, this paper explores ways to make the best of prior resources and improve the effectiveness of unsupervised language analysis techniques for the purpose of linguistic documentation. Our main objective is to develop tools that will effectively assist field linguists in their documentary tasks (objective (a)). We focus on segmentation tasks, which aim to automatically identify meaningful units in an unsegmented phonetic or orthographic string (Johnson, 2008; Doyle and Levy, 2013; Eskander et al., 2016; Godard et al., 2018b; Eskander et al., 2019).

Following these authors, we experiment with Bayesian non-parametric segmentation models, derived in our case from Goldwater et al. (2009) and subsequent work, which we recap in Section 2. Our first contribution is in Section 3 which studies multiple semi-supervised learning regimes aimed to take advantage of pre-existing linguistic material such as incomplete segmentations and word lists.

In Sections 4 and 5, we experimentally assess the pros and cons of these weakly supervised approaches in batch and online learning, for two extremely low-resource languages currently in the process of being documented: Mboshi, a Bantu language used in former studies (Godard et al., 2018a); and Japhug, a language from the Sino-Tibetan family spoken in the Western part of China thoroughly documented by Jacques (2021). These two languages were selected because they illustrate actual documentation processes, for which high-quality linguistic resources have been derived from fieldwork, at the end of a long and difficult procedure (Aiton, 2021). A complementary analysis follows,

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7385 - 7398

where we use the Japhug corpus to take a closer look at the units identified automatically, contrasting morpheme-based and word-based supervision.

2 Background

Going from audio recordings to fully annotated transcripts implies two successive segmentation steps: the first segments words and happens during the production of phonemic or orthographic transcripts; the second further splits words into morphs, which are then annotated with syntactic information and glosses. We mostly focus on the former task, assuming a two-step process: first, the computation of a phonemic transcript that we assume is given; then the segmentation into words for which we consider two settings: batch and online learning. The word and morpheme segmentation tasks are closely related and rely on similar tools: using the Japhug corpus, which contains both levels of segmentations, we also study the implications of using lists of words vs morphemes as weak supervision.

In its baseline form, the word segmentation process is fully unsupervised, and the only training material is a set of transcribed sentences (see Fig. 1).

We rely on Bayesian non-parametric approaches to word segmentation (see (Cohen, 2016) for a thorough exposition), and our baselines are the unigram version of the `dpseg` model (Goldwater et al., 2009) and a variant where the underlying Dirichlet Process is replaced by a Pitman-Yor Process as in (Neubig, 2014). We selected unigram models for their simplicity, which (a) makes them amenable to the processing of very small sets of sentences; (b) makes the online learning setting tractable. While using higher-order models or more sophisticated models of the same family (Teh, 2006b; Mochihashi et al., 2009) may improve the performance (see (Godard et al., 2016) for an experimental comparison), we believe that in our low-resource conditions, these variations would be small[2] and would not change our main conclusions.

Word segmentation models fundamentally rely on probabilistic models for word sequences defining $P(\boldsymbol{w} = w_1 \dots w_T)$; word sequences can also be viewed as segmented sequences of characters $\boldsymbol{y} = y_1 \dots y_L$, so that the same model can be used for the joint probability of $(\boldsymbol{y}, \boldsymbol{b})$, with $\boldsymbol{b} = b_1 \dots b_L$ representing the vector of boundary locations where value $b_t = 1$ (resp. $b_t = 0$) denotes a boundary (resp. no boundary) after symbol y_t. In an unsupervised setting, these boundaries are hidden and are latent variables in the model. Such models lend themselves well to Gibbs sampling, which repeatedly produces samples of each boundary given all the other boundaries in the corpus.

In `dpseg`, the underlying sequence model is a unigram model: $P(w_1 \dots w_T) = \prod_{t=1}^{T} P(w_t)$. The probability of individual words corresponds to a Dirichlet Process with parameters α, the *concentration* parameter, and P_0, the *base distribution*, and yields the following formulation for the conditional probability of w_t given the past words $\boldsymbol{w}_{<t}$:

$$P(w_t = w | \boldsymbol{w}_{<\boldsymbol{t}}) = \frac{n_w(\boldsymbol{w}_{<\boldsymbol{t}}) + \alpha P_0(w)}{t + \alpha - 1}, \quad (1)$$

where $n_w(\boldsymbol{w}_{<\boldsymbol{t}})$ counts the number of times w has occurred in the past. With lower values of α, the most frequent words tend to be generated more (hence, concentration), while with higher values, the words are more smoothly distributed. P_0, the base distribution, assigns scores to arbitrary character strings; Goldwater et al. (2009) use a length model and a uniform character model. For word w made of characters $y_1, ..., y_m$, P_0 is computed as:

$$P_0(w) = \underbrace{p_\#(1-p_\#)^{m-1}}_{\text{length model}} \underbrace{\prod_{j=1}^{m} P(y_j)}_{\text{character model}} \quad (2)$$

where $p_\#$ is the probability to end the word.

For this model, Gibbs sampling compares at each position t two sequences of words $\boldsymbol{w}_{t=0}$ (no boundary at position t) and $\boldsymbol{w}_{t=1}$ (a boundary is inserted). As these sequences only differ minimally, terms such as $P(b_t = 0 | \boldsymbol{y}, \boldsymbol{b}_{-t})$ are readily derived (see e.g. (Goldwater et al., 2009)). Gibbs sampling is performed for a number of iterations that are sufficient to reach convergence, and we use the last iteration to uncover the resulting segmentation. To speed up mixing, Goldwater et al. (2009) also use annealing, so that a larger search space is explored.

An extension of `dpseg`, denoted `pypseg`, uses a Pitman-Yor Process (PYP) instead of the Dirichlet Process and generalises equation (1) with an additional *discount* parameter, which enables to better control the generation of new words. PYPs are introduced in (Teh, 2006b; Mochihashi et al., 2009); a fast implementation is in (Neubig, 2014). For our experiments, both models have

[2] Godard et al. (2018a) report results with the bigram version of dpseg on the Mboshi corpus; the difference with our unigram version is about 4 points for the boundary F-score.

$\boldsymbol{y} =$	b_1	$á_2$	a_3	$á_4$	m_5	i_6	k_7	$ú_8$	n_9	d_{10}	$á_{11}$	p_{12}	o_{13}	o_{14}	y_{15}	$á_{16}$	k_{17}	a_{18}	l_{19}	a_{20}
$\boldsymbol{b} =$	0	0	1	0	0	**0**	0	0	0	0	1	0	0	1	0	1	0	0	0	1
$\boldsymbol{w}_{6=0}$		báa				ámikúndá							poo			yá			kala	
$\boldsymbol{b} =$	0	0	1	0	0	**1**	0	0	0	0	1	0	0	1	0	1	0	0	0	1
$\boldsymbol{w}_{6=1}$		báa			ámi				kúndá				poo			yá			kala	

Figure 1: The sentence segmentation task illustrated with a sentence from the Mboshi corpus: 'báa ámikúndá poo yá kala' ('*they found the old village*'). The two possible segmentations only differ in one boundary at position $t = 6$, one ($w_{6=0}$) where 'ámikúndá' is one single unit and one ($w_{6=1}$) where it is split in two.

been re-implemented in Python. This implementation is available at `https://github.com/shuokabe/pyseg`.

3 Supervising word segmentation

In this section, we discuss realistic sources of weak supervision for segmentation tasks and how they can be included in Bayesian models.

3.1 Finding supervision information

Segmentation boundaries Segmentation data, corresponding to the location of boundary (and non-boundary) information, can be obtained in different ways. For instance, when audio recordings are available, prosodic cues such as short silences or specific intonative patterns can serve to identify plausible locations for word endings. Longer pauses generally denote the end of an utterance, which we assume are already given. This would yield a *sparse partial annotation*, where supervision data is randomly scattered across the corpus.

Another realistic situation where we have access to a partial annotation is when a small subset is already segmented. In this case, the partial annotation is *dense* and concentrated in a few sentences, a semi-supervised setting also studied in (Sirts and Goldwater, 2013). We thus consider two questions: (a) which is more effective between dense and sparse annotations? (b) how effective is supervision in an incremental learning regime, where automatic (dense) annotations are progressively corrected and used to update the model?

Word lists Word lists constitute another valuable and common source of information. They may contain morphs, morphemes, lexemes or fully inflected forms, with various levels of information (part-of-speech, gloss, translation, etc.). In this study, we consider that lists of *surface forms* are available and evaluate their usefulness, depending on their size and on the way they were collected. A related question is about the relative interest of word and morph lists, which we study in Section 5.3. The use of more sophisticated forms of lexical information regarding word structure, PoS, is out of the scope of this paper and is left for future work.

Having a collection of fully segmented utterances, as discussed above, is another way to generate word lists. So these two sources of information must be viewed as complementary ways to supervise the task at hand: boundary marks at the token level, word list at the type level.

3.2 Forms of Weak Supervision

Segmentation boundaries Observed segmentation boundaries can be used to facilitate the training process. Two experimental conditions, both affecting the Gibbs sampler (`gs`), have been considered:

- `gs.sparse`: a fraction ($\lambda\%$) of the actual boundaries are observed, which corresponds to a sparse annotation scenario.
- `gs.dense`: for $\lambda\%$ of sentences, all boundary and non-boundary variables are given.

In both cases, we modify the sampling process and make sure that the value of observed variables is not sampled, as in (Sirts and Goldwater, 2013).

Using a word list Assuming now that a word list D is available, we consider the following approaches to reinforce the likelihood of units in D in the output segmentation:

- `d.count`: D is used to initialise the 'internal' model dictionary, and words in D are created with a fixed pseudo-count of value λ. Formally, $\forall w \in D$, the counting function $n_w()$ of Equation (1) will add λ to their actual count.
- `d.mix`: D is combined with the base distribution, resulting in the following mixture P_0':

$$P_0'(w) = \frac{\lambda}{|D|}\mathbb{1}_{\{w \in D\}} + (1 - \lambda)P_0(w), \quad (3)$$

where $\lambda \in [0, 1]$, $|D|$ is the size of D, and $\mathbb{1}_{\{w \in D\}}$ is the indicator function testing membership in D. As for `d.count`, P_0' increases the probability of words in D, but in a looser way, due to the term αP_0 in Equation (1).

- `d.ngram`: the baseline `dpseg` version uses a uniform character model for P_0 (Equation (2)); here, we use D to train a character n-gram language model (LM), with $n = 2$ and add-k smoothing in our experiments.

- `d.mix+ngram`: this method combines `d.mix` and `d.ngram`: P_0 is replaced with the mixture P_0' of Equation (3) and the character model is an n-gram LM. This can be viewed as a proxy to the complete nested Dirichlet Process of Mochihashi et al. (2009), with D implementing a cache mechanism for known words.

We have also used weaker forms of supervision aimed at learning a better length model, with hardly any improvement with respect to the baseline; these results are not reported below.

3.3 Incremental training

In addition to the static use of supervision information described above, we also considered a more dynamic training regime, where dense annotations are provided in a sequential manner through interaction with an expert linguist, enabling incremental learning. To measure the effectiveness of this approach, we contrast three scenarios in Section 5.2:

- the baseline is the post-edition of a fully unsupervised model without further training;

- the post-edition of a fully unsupervised model, with additional Gibbs sampling iterations every batch utterances for iter iterations. This aims at propagating forward the supervision information obtained from past annotations. This method is referred to as `o.regular`.

- on top of this, we also used the past annotated sentences to reestimate the base distribution of the underlying process as in `d.ngram`. The corresponding results are labelled `o.2level` in Figure 2.

4 Experimental settings

4.1 Linguistic material

Two languages have been considered in this paper: Mboshi and Japhug.

Mboshi is a tonal Bantu language spoken in the Republic of Congo (Bantu C25). The data has been collected as part of the BULB project (Adda et al., 2016). It has seven vowels and 25 consonant phonemes with five prenasalised consonants (made of two to three consonants), a common feature in Bantu languages (Embanga Aborobongui, 2013; Kouarata, 2014). Although the language is usually not written, linguists have transcribed it with graphemes in a way that approximates the phonetic content. To mark the distinction between long and short vowels, they were either duplicated (VV) or not (V). One challenge for Mboshi word segmentation is its complex phonological rules, notably, vowel elision patterns whereby a vowel disappears before another one (also a common Bantu feature) (Rialland et al., 2015). This kind of phenomenon makes it harder to find the boundaries.

From a morphological point of view, words are composed of roots and affixes. Another characteristic Bantu feature is its deletion rule for class-prefix consonants in nouns. Templates for verb structure are also quite rigid, with affixes following a strict ordering (Godard et al., 2018a).

Our corpus is a manual alphabetic transcription of audio recordings.[3] It contains 5,312 sentences segmented in words, one sentence per line.

Japhug is a Sino-Tibetan language from the Gyalrong family spoken in the Sichuan province in China. Japhug has eight vowels and 50 consonant phonemes, which can combine to create a large number (more than 400) of consonant clusters. The rich cluster feature is one important characteristic of Japhug, which actually has one of the largest inventory of consonant clusters in the Trans-Himalayan language family. The structure of these clusters can be analysed by looking at patterns of partial reduplication of syllable initial consonants. There are no tones in this language.

Japhug also has a rich morphology, both for verbs and nouns. Remarkably, in verb forms, up to six or seven prefixes can be chained to express features such as tense, aspect, modality, while suffixation is used to express inflectional phenomena.

[3]Download from: https://www.islrn.org/resources/747-055-093-447-8/.

Even though these processes are quite regular, they contribute to generating a large number of possible word forms. Recordings, annotated corpora, and dictionaries for Japhug are available from the Pangloss collection.[4] An extensive description of the language is given in (Jacques, 2021).[5]

Our training material has been extracted from the LaTeX source files of this book, by collecting all Japhug examples. These can easily be retrieved by searching the `\gll` command introducing Japhug sentences. Not only are the resulting sentences well-curated, but they are also segmented at two levels: words and morphemes. This will lead to a specific experiment presented in Section 5.3.

language	Mboshi	Japhug	
segment	word	word	morph.
N_{utt}	5130	3628	3628
WL	4.19	4.73	2.90
TL	6.39	7.30	5.41
N_{type}	5312	6739	2731
N_{token}	30556	28579	46632

Table 1: Statistics for the Mboshi and Japhug corpora. For the latter, we use the word-based and morpheme-based segmentations.

Table 1 displays the general statistics for the two languages. N_{utt}, N_{type}, and N_{token} represent the number of utterances, of word types, and of word tokens, respectively. WL represents the average token length, while TL is the average type length. The sentences used for semi-supervision correspond to the first 200 sentences of each dataset, which is a realistic amount of data. Likewise, lexical supervision corresponds to the list of words observed in the same 200 sentences, and respectively contain 517 words for Mboshi, 664 words and 493 morphemes for Japhug.

4.2 Model settings

In our experimental setting, we made sure to also resample the hyperparameter(s) after each iteration, following mostly (Teh, 2006a; Mochihashi et al., 2009): the concentration parameter α has a Gamma posterior distribution, and the discount parameter d a Beta distribution. The initial values of the hyperparameters were set as in Goldwater et al.'s work on the unigram `dpseg`: concentration parameter: $\alpha = 20$, $p_{\#} = 0.5$, discount parameter for `pypseg`: $d = 0.5$. The Gibbs sampler always runs for 20,000 iterations and simulated annealing is implemented as in (Goldwater et al., 2009) with 10 increments of temperature.

All the results are obtained by collecting the predicted boundaries at the end of the last sampling iteration of one single run.

4.3 Evaluation metrics

Following Goldwater et al. (2009), evaluation relies on 'PRF' metrics: precision, recall, and F-score, defined as follows: precision $P = \frac{TP}{TP+FP}$, recall $R = \frac{TP}{TP+FN}$, and F-score $F = 2 * \frac{precision * recall}{precision + recall}$, where TP are the true positives (match in the reference and segmented texts), FP are the false positives, and FN are the false negatives. These metrics are computed at three levels:[6]

- boundary level (BP, BR, BF): compare the reference boundary vectors with the predictions;
- token level (WP, WR, WF): compare word in the reference and segmented sentences: a correct match requires two correct boundaries;
- type level (LP, LR, LF): compare the set of unique words in the reference and segmented utterances.

To have an overall view of the output text, we also report the average type and token lengths (TL and WL) as well as their counts (N_{type} and N_{token}), as in Table 1. Numbers are computed on the entire text (including the supervised part).

5 Results

This section presents the results for the models presented above. We also report the performance of SentencePiece, another word segmentation tool based on a unigram language model (Kudo, 2018):[7] To boost this baseline, the vocabulary size has been set to the *reference number* of N_{type} (cf. Table 1). Supplementary material additionally contains results for Morfessor baselines (Creutz and Lagus, 2002), with the corresponding weak supervision. As a reminder, our supervision here consists of the first 200 sentences in the text, either directly given as observed boundaries or used to generate the initial word list.

[4] http://pangloss.cnrs.fr/corpus/Japhug.

[5] Available at https://github.com/langsci/295/tree/main/chapters.

[6] Below we only report F-scores; complete results are in the appendix A.1.

[7] github.com/google/sentencepiece.

5.1 Using weak supervision

5.1.1 dpseg

Table 2 displays our experimental results for the 5K Mboshi corpus for SentencePiece (SP), `dpseg` and `pypseg` with various amounts of supervision.

First, the unsupervised `dpseg` model has better results than SP on all three levels by a significant margin. SP, on the other hand, produces more types as it 'knows' the actual number of types to generate.

Regarding segmentation boundaries, the `gs.sparse` model has disappointing results, with scores lower than the baseline. On the other hand, the dense supervision manages to improve the baseline scores by around 2.5 points for BF, 4.5 points for WF, and 7.5 points for LF. This is an encouraging result, since, with less than 5% of the whole text, the model has improved in a noticeable way, especially at type level, which seems to be difficult for fully unsupervised learning.

When supervising with a word list, all models but `d.2gram` outperform the baseline. Yet, the `d.count` and `d.mix` methods have lower scores than the `gs.dense`: this was expected for BF and WF—where directly supervising boundaries is likely to be more useful than an indirect one, but less so for LF. Regarding the `d.2gram` model, its poor BF and WF scores are more than compensated by an increase of around 12 points in LF, showing the impact of a better type model. Finally, by combining the `d.mix` and `d.2gram` strategies, `d.mix+2gram` obtains the overall best results.

5.1.2 pypseg

Results are in the right part of Table 2, where the baseline is the fully unsupervised `pypseg`. It slightly outperforms `dpseg` by less than 1 point in terms of F-scores. In our setting, although PYP increases the number of discovered types, it does not improve the performance in any significant manner.

This trend is confirmed for weakly supervised models:[8] the `gs.dense` model is the only one benefiting from a small improvement in all F-scores. `d.count` underperforms both the baseline and its `dpseg` version. With worsened BF and WF scores compared to the baseline, `d.mix+2gram` with `pypseg` is worse than with `dpseg`. Overall, the former seems to benefit less from annotations than the latter.

[8] We do not report the results of `d.mix` and `d.2gram` but their combination `d.mix+2gram`, due to space limitation.

The performance of the bigram character model is noteworthy both with `dpseg` and `pypseg`. This improvement alone (i.e. `d.2gram`) is responsible not only for a large increase in LF, but also for an average type length that gets much closer to its true value (6.39 in the reference, 6.60 with `dpseg` and `d.mix+2gram`).

5.1.3 Results for Japhug

Table 3 displays a selection of results for Japhug (segmented in words). As previously observed, supervision noticeably improves the results for both models, with `pypseg` outperforming `dpseg` by a small margin on all metrics.[9] Note also that SP is much worse than Bayesian models, only reaching the same F-score as `dpseg` for the LF metric.

The best results are obtained with lexical supervision and the `d.mix+2gram` model for `dpseg`: it combines the type boost in P_0' from `d.mix` and the improved base model from `d.2gram`.

5.2 Incremental learning

Figure 2 displays the evolution of the boundary error rate (number of errors over 100 sentences / length of the 100 sentences) as more annotated sentences are available, for three contrasts of § 3.3 (baseline, `o.regular`, and `o.2level`). We use the `dpseg` model and 50 complementary Gibbs sampling iterations every 100 sentences.

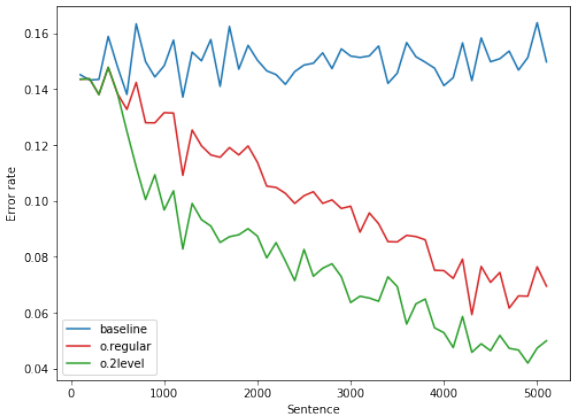

Figure 2: Average loss over 100 sentences on the 5K Mboshi text with incremental training (batch = 100, iter = 50)

While the baseline error rate (in blue) remains the same throughout training, both supervised models show a sharp decrease, from 0.14 to about 0.06. The large drop at the beginning for the `o.2level`

[9] Full results are in appendix A.1.

mod.	SP	dpseg							pypseg			
sup.	/	base.	sparse	dense	count	mix	2gram	mix+2	base.	dense	count	mix+2
BF	44.6	65.9	65.0	68.7	66.3	68.3	64.7	66.4	66.2	68.8	65.5	65.8
WF	17.7	37.6	36.4	42.4	38.1	41.7	36.5	39.4	37.9	42.5	37.6	38.7
LF	19.5	23.8	22.0	31.4	23.9	30.7	36.1	40.0	24.5	31.6	24.0	39.9
WL	3.89	3.74	3.50	3.78	3.73	3.79	5.10	5.11	3.77	3.82	3.80	5.16
TL	6.93	4.61	4.45	4.87	4.60	4.87	6.57	6.60	4.65	4.89	4.79	6.62
N_{type}	5031	1980	1938	2237	1999	2181	4636	4620	2063	2310	2163	4741
$N_{tok.}$	32.9k	34.2k	36.6k	33.8k	34.3k	33.8k	25.1k	25.0k	33.9k	33.5k	33.7k	24.8k

Table 2: Results on the 5K Mboshi text for various models and weak supervision settings (20K iterations, 200 supervision sentences, $\lambda = 0.25$). SP stands for SentencePiece; mix+2 for `d.mix+2gram`.

mod.	SP	dpseg			pypseg
sup.	/	base.	dense	mix+2	base.
BF	59.7	72.9	75.0	78.8	73.0
WF	30.3	45.7	50.4	55.8	46.1
LF	20.0	20.1	28.3	42.7	20.8
WL	4.72	3.34	3.44	4.50	3.36
TL	6.71	4.21	4.67	6.19	4.25
N_{type}	6413	2258	2610	5041	2295
$N_{tok.}$	28.6k	40.5k	39.3k	30.0k	40.2k

Table 3: Results on the 3K Japhug text with various models (20K iterations, 200 supervision sentences). SP stands for SentencePiece; mix+2 for `d.mix+2gram`.

ref.	word			morpheme		
sup.	/	word	mor.	/	word	mor.
BF	72.9	78.8	76.1	80.8	71.0	75.8
WF	45.7	55.8	51.0	54.7	39.2	45.1
LF	20.1	42.7	32.8	41.2	33.5	43.8
WL	3.34	4.50	4.09	3.34	4.50	4.09
TL	4.21	6.19	5.43	4.21	6.19	5.43
N_{type}	2258	5041	4077	2258	5041	4077
$N_{tok.}$	40.5k	30.0k	33.1k	40.5k	30.0k	33.1k

Table 4: Comparison of the results on the 3K Japhug text with the word or morpheme segmented reference (ref.), dictionary from 200 supervision sentences (sup.)

model (green) can be attributed to the use of the bigram character model. It gives this model an initial edge over `o.regular` that remains significant for the first 3,000 sentences. Here again, the benefits of improving the base distribution (character-based model) as much as possible in the early training iterations clearly appear.

5.3 Supervising words and morphemes

This section addresses a recurring issue in word segmentation model related to the linguistic nature of the units learnt by the model and the consequences of choosing one or the other reference in training. The Japhug corpus contains both annotation levels and is a perfect test bed for this study. We have thus used a segmentation model (`dpseg`) with and without weak supervision (using the `d.mix+2gram` variant) at the level of words or morphemes, and the results are also evaluated against the two references (a segmentation in words or in morphemes). Results are in Table 4.

In the unsupervised setting, segmentation metrics are markedly better with morpheme-based references, especially for the LF metric. This again shows the tendency of the unigram model to oversegment the training sentences.

With word supervision, we observe a shift in behaviour that is consistent with the provided annotations: better word-level metrics with word-based annotations, and accordingly, a decrease of performance for morpheme-based scores. With morpheme supervision, results are more contrasted: an improvement for word segmentation (because some words are also morphemes) that is not matched for morpheme boundaries. Looking at the detailed results (see appendix A.1, Table 7), one can see that this is due to an undersegmentation, which yields a poor recall at the boundary and token levels. Here, the main remaining benefit of supervision is an increase in the LF score.

These preliminary results suggest that considering only one type of boundary is a too naive view of the segmentation process and does not allow us to fully benefit from annotated data. They call for models that would carefully distinguish boundaries within words and between words, with appropriate supervision for each of these levels.

5.4 Error analysis

It is noteworthy that dictionary supervision almost deterministically ensures that the input word types will occur in the segmented output. For instance, 96% of the words in the Mboshi supervision dictionary are found in the output of the `d.mix+2gram` method, whereas we only find 44% with fully unsupervised learning. Similar trends are observed for Japhug. Some remaining errors are, however, observed: in the example of Figure 3, the word 'bana' belongs to the supervision dictionary but remains attached to the following word 'ba'. Additional examples are in appendix A.2. This may be because both words 'bana' and 'ba' often occur together, a cooccurrence that can not be captured by our unigram model (Goldwater et al., 2009).

reference	**bana** ba adi otɛɛ imbva
unsupervised	**bana**ba adio tɛɛ imbva
supervised	**bana**ba adi otɛɛimbva

Figure 3: Example of a segmentation error for the Mboshi sentence: '*these children are the same size*'.

6 Related work

Unsupervised segmentation is a generic NLP task that can be performed at multiple levels of analysis: a document segmented in sections, a speech segmented in utterances, an utterance segmented in words, a word segmented in morphemes, syllables or phonemes. It has been studied in multiple ways, and we report here recent work related to word discovery for language documentation, noting that the same methods also apply to the unsupervised segmentation of continuous speech into 'words' (de Marcken, 1996) which has given rise to a vast literature on language acquisition. Recently, this task has become central in preprocessing pipelines, with new implementations of simple models (Sennrich et al., 2016; Kudo and Richardson, 2018).

Linear segmentation models in the Bayesian realm can be traced back to (Goldwater et al., 2006, 2009). They were extended with nesting in (Mochihashi et al., 2009), where the base distribution of the Dirichlet Process is a char-based non-parametric model; and in (Uchiumi et al., 2015; Löser and Allauzen, 2016), who consider hidden state variables in the word generation process. This extension enables, for instance, to jointly learn segmentation and PoS tagging or to introduce some morphotactics in the model. Other sources of weak supervisions along these lines concern the use of higher-order n-grams and of prosodic cues (Doyle and Levy, 2013). Finally, (Börschinger and Johnson, 2012) (with particle filtering techniques) and (Neubig, 2014) (with block sampling) study ways to speed up inference.

The unsupervised techniques exposed in Section 2 only depend on the design of a probabilistic word generation process. This means that they are also readily applicable when this process is conditioned to some input, for instance, when a translation is available as an additional information source. This setup is notably studied in (Neubig et al., 2011; Stahlberg et al., 2012), and also considered, with radically different tools, in (Anastasopoulos and Chiang, 2017; Godard et al., 2018c).

A somewhat richer trend of works aimed at informing word segmentation relies on the model of adaptor grammars (AG) of Johnson et al. (2007), applied to the segmentation task as early as (Johnson, 2008). AGs generalise finite-state models such as `dpseg` and `pypseg` by modelling trees and subtrees, rather than mere strings. Their use necessitates a context-free description of the language, which enables to integrate information regarding word and syllable structures. Even generic descriptions can be useful, but finding the most appropriate and effective one is challenging (Johnson and Goldwater, 2009; Eskander et al., 2016). This formalism has also been used to introduce syntactic information (Johnson et al., 2014), prosodic information (Börschinger and Johnson, 2014), and partial annotations (Sirts and Goldwater, 2013). Recent software packages for AGs are presented in (Bernard et al., 2020) and (Eskander et al., 2020). Using AGs comes, however, with a high computational price, as the Gibbs sampling process typically requires repeated parses of the corpus, even though cheaper estimation techniques may also be considered (Cohen et al., 2010). As our goal is to integrate learning techniques in interactive annotation tools, AGs were not deemed appropriate, and we explored simpler alternatives.

Similar arguments apply to the use of neural networks, which have attracted a growing interest even for very low-resource languages, combining supervised segmentation methods (Moeng et al., 2021; Liu et al., 2021) with cross-lingual transfer or data augmentation techniques (Silfverberg et al., 2017; Kann et al., 2018; Lane and Bird, 2020).

7 Conclusion and outlook

In this work, we have studied various ways to use weak supervision for automatic word segmentation. In language documentation scenarios, such supervision is often available, taking the form of a partial annotation or word lists. Bayesian non-parametric models lend themselves well to this setting, and our experiments have shown that two variants of a simple unigram model were getting a substantial boost from weak supervision, a result that has been obtained with two languages currently being documented. The most effective approach seems to start with a small set of fully segmented data, which helps learning in two ways: as a training signal for segmentation and as lexical prior for the base distribution. Based on this observation, we have further evaluated the longer-term benefits of an incremental training regime and also contrasted the improvement obtained using a word-based vs a morpheme-based vocabulary list.

Our future work will continue to explore the interplay between word and morpheme segmentations, as both are required in actual documentation settings, possibly extending our analyses on additional languages. We will also consider supervising the annotation process with lists of *non-inflected forms*, which requires to jointly learn inflectional patterns and segmentation. Finally, our main objective remains to integrate these techniques into an annotation platform and evaluate how much they help speed up the annotation process, hence the need to control the run-time of our algorithms.

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 Alexis Michaud and Guillaume Jacques for their help in preparing the Japhug corpus.

References

Gilles Adda, Sebastian Stüker, Martine Adda-Decker, Odette Ambouroue, Laurent Besacier, David Blachon, Hélène Bonneau-Maynard, Pierre Godard, Fatima Hamlaoui, Dmitri Idiatov, Guy-Noël Kouarata, Lori Lamel, Emmanuel-Moselly Makasso, Annie Rialland, Mark Van de Velde, François Yvon, and Sabine Zerbian. 2016. Breaking the Unwritten Language Barrier: The Bulb Project. In *Proceedings of SLTU (Spoken Language Technologies for Under-Resourced Languages)*, Yogyakarta, Indonesia.

Grant Aiton. 2021. Translating fieldwork into datasets: The development of a corpus for the quantitative investigation of grammatical phenomena in Eibela. *Proceedings of the Workshop on Computational Methods for Endangered Languages*, 2(2).

Antonios Anastasopoulos and David Chiang. 2017. A case study on using speech-to-translation alignments for language documentation. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 170–178, Honolulu. Association for Computational Linguistics.

Mathieu Bernard, Roland Thiolliere, Amanda Saksida, Georgia R. Loukatou, Elin Larsen, Mark Johnson, Laia Fibla, Emmanuel Dupoux, Robert Daland, Xuan Nga Cao, and Alejandrina Cristia. 2020. Wordseg: Standardizing unsupervised word form segmentation from text. *Behavior Research Methods*, 52(1):264–278.

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.

Benjamin Börschinger and Mark Johnson. 2012. Using rejuvenation to improve particle filtering for bayesian word segmentation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–89, Jeju Island, Korea. Association for Computational Linguistics.

Benjamin Börschinger and Mark Johnson. 2014. Exploring the role of stress in Bayesian word segmentation using Adaptor Grammars. *Transactions of the Association for Computational Linguistics*, 2:93–104.

Shay Cohen. 2016. *Bayesian Analysis in Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Shay B. Cohen, David M. Blei, and Noah A. Smith. 2010. Variational inference for Adaptor Grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 564–572, Los Angeles, 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.

Carl de Marcken. 1996. *Unsupervised Language Acquisition*. Ph.D. thesis, Department of Computer Science, MIT.

Gabriel Doyle and Roger Levy. 2013. Combining multiple information types in Bayesian word segmentation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 117–126, Atlanta, Georgia. Association for Computational Linguistics.

Georges Martial Embanga Aborobongui. 2013. *Processus segmentaux et tonals en Mbondzi - (variété de la langue embosi C25) -*. Theses, Université de la Sorbonne nouvelle - Paris III.

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, Gilles Adda, Martine Adda-Decker, Juan Benjumea, Laurent Besacier, Jamison Cooper-Leavitt, Guy-Noel Kouarata, Lori Lamel, Hélène Maynard, Markus Mueller, Annie Rialland, Sebastian Stueker, François Yvon, and Marcely Zanon-Boito. 2018a. A very low resource language speech corpus for computational language documentation experiments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Pierre Godard, Laurent Besacier, François Yvon, Martine Adda-Decker, Gilles Adda, Hélène Maynard, and Annie Rialland. 2018b. 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.

Pierre Godard, Marcely Zanon Boito, Lucas Ondel, Alexandre Berard, François Yvon, Aline Villavicencio, and Laurent Besacier. 2018c. Unsupervised word segmentation from speech with attention. In *Proc. Interspeech 2018*, pages 2678–2682, Hyderabad, India.

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.

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, Anne Christophe, Emmanuel Dupoux, and Katherine Demuth. 2014. Modelling function words improves unsupervised word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 282–292, Baltimore, Maryland. Association for Computational Linguistics.

Mark Johnson and Sharon Goldwater. 2009. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 317–325, Boulder, Colorado. 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.

Guy Noël Kouarata. 2014. *Variations de formes dans la langue Mbochi (Bantu C25)*. Ph.D. thesis, Université Lumière Lyon 2.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

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.

William Lane and Steven Bird. 2020. Bootstrapping techniques for polysynthetic morphological analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6652–6661, Online. 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.

Kevin Löser and Alexandre Allauzen. 2016. Une méthode non-supervisée pour la segmentation morphologique et l'apprentissage de morphotactique à l'aide de processus de Pitman-Yor (an unsupervised method for joint morphological segmentation and morphotactics learning using Pitman-Yor processes). In *Actes de la conférence conjointe JEP-TALN-RECITAL 2016. volume 2 : TALN (Articles longs)*, pages 207–220, Paris, France. AFCP - ATALA.

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: Volume 1-Volume 1*, pages 100–108. 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.

Graham Neubig. 2014. Simple, correct parallelization for blocked Gibbs sampling. Technical report, Nara Institute of Science and Technology.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 632–641.

Annie Rialland, Georges Aborobongui, Martine Adda-Decker, and Lori Lamel. 2015. Dropping of the class-prefix consonant, vowel elision and automatic phonological mining in Embosi (Bantu C 25). In *Proceedings of the 44th Annual Conference on African Linguistics*, pages 221–230, Somerville. Cascadilla.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver. Association for Computational Linguistics.

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.

Felix Stahlberg, Tim Schlippe, Stephan Vogel, and Tanja Schultz. 2012. Word segmentation through cross-lingual word-to-phoneme alignment. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 85–90. IEEE.

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

Yee Whye Teh. 2006b. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992, Sydney, Australia. Association for Computational Linguistics.

Kei Uchiumi, Hiroshi Tsukahara, and Daichi Mochihashi. 2015. Inducing word and part-of-speech with Pitman-Yor hidden semi-Markov models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1774–1782, Beijing, China. Association for Computational Linguistics.

A Appendix

A.1 Full results

Table 5 displays the complete results of Table 2 with both precision and recall for the three evaluation levels. SentencePiece (SP) tends to have more balanced scores for precision and recall, whereas `dpseg` displays a wider gap between the two metrics, especially at type level.

The 'Morf' column displays the performance of Morfessor 2.0 (Creutz and Lagus, 2002; Smit et al., 2014).[10] These results have been obtained with the morph-length parameter set to the observed average token length (4.19). This setting led to better F-scores than using the gold number of types for num-morph-types or the default Morfessor model. The Morfessor model outperforms SentencePiece significantly for both boundary (BF) and token (WF) F-scores, while it lags behind for the type-based metrics. Compared to the unsupervised `dpseg`, Morfessor is worse on all accounts by a wide margin.

Table 6, in turn, displays the complete results of Table 3, again with both precision and recall for the three evaluation levels.

The 'Morf' column in Table 6 also represents the Morfessor results, with a morph-length parameter of 4.73. Here again, Morfessor outperforms SentencePiece on the boundary and token-level F-scores (to a smaller extent) but not at type level.

Finally, Table 7 displays the complete results for the word and morpheme experiment (Table 4).

A.2 Output analysis

reference	**obengi** amipasa koo sa kω
unsupervised	**obengi**a mipasa koo sakω
supervised	**obengi** amipasa koo sakω

Figure 4: Example of Mboshi sentence ('*the hunter made a path through the forest*') corrected through supervision (here, `d.mix+2gram`): 'obengi' is a word in the supervision dictionary

Figure 4 shows an example sentence derived from the Mboshi data. The word 'obengi' is present in the supervision dictionary. In the unsupervised model (unsupervised line), the word was wrongly

[10] https://github.com/aalto-speech/morfessor.

mod.	SP	Morf	dpseg							pypseg			
sup.	/	/	base.	sparse	dense	count	mix	2gram	mix+2	base.	dense	count	mix+2
BP	42.71	55.76	61.79	58.83	64.80	62.09	64.46	73.57	75.63	62.30	65.20	61.88	75.51
BR	46.65	53.85	70.66	72.73	73.09	71.15	72.58	57.68	59.15	70.51	72.78	69.51	58.36
BF	44.59	54.78	65.93	65.05	68.70	66.31	68.28	64.66	66.39	66.15	68.78	65.48	65.84
WP	17.07	29.43	35.63	33.45	40.41	36.03	39.68	40.47	43.82	36.07	40.58	35.81	43.20
WR	18.38	28.59	39.88	40.03	44.71	40.40	43.83	33.19	35.87	40.03	44.51	39.48	35.03
WF	17.70	29.00	37.63	36.45	42.45	38.09	41.65	36.47	39.45	37.95	42.45	37.56	38.69
LP	20.00	21.22	43.84	41.23	53.06	43.72	52.82	38.74	43.03	43.72	52.12	41.42	42.27
LR	18.94	10.64	16.34	15.04	22.35	16.45	21.69	33.81	37.42	16.98	22.67	16.87	37.73
LF	19.45	14.17	23.81	22.04	31.45	23.91	30.75	36.11	40.03	24.46	31.59	23.97	39.87
WL	3.89	4.31	3.74	3.50	3.78	3.73	3.79	5.10	5.11	3.77	3.82	3.80	5.16
TL	6.93	8.91	4.61	4.45	4.87	4.60	4.87	6.57	6.60	4.65	4.89	4.79	6.62
N_{type}	5031	2663	1980	1938	2237	1999	2181	4636	4620	2063	2310	2163	4741
$N_{tok.}$	32901	29685	34204	36562	33810	34264	33755	25063	25015	33905	33514	33691	24782

Table 5: Complete results on the 5K Mboshi text for various models and weak supervision settings (20K iterations, 200 supervision sentences, $\lambda = 0.25$). SP stands for SentencePiece, Morf for Morfessor.

mod.	SP	Morf	dpseg			pypseg	
sup.	/	/	base.	dense	mix+2	base.	mix+2
BP	59.65	53.43	61.10	63.75	76.62	61.39	76.67
BR	59.82	74.93	90.20	91.19	81.11	90.08	80.22
BF	59.74	62.38	72.85	75.04	78.80	73.02	78.40
WP	30.28	31.12	38.98	43.55	54.44	39.42	53.97
WR	30.35	42.05	55.18	59.92	57.23	55.51	56.15
WF	30.31	35.77	45.69	50.44	55.80	46.10	55.04
LP	20.51	20.05	39.95	50.77	49.93	40.92	49.32
LR	19.51	8.00	13.38	19.66	37.35	13.93	37.50
LF	20.00	11.44	20.05	28.35	42.73	20.79	42.60
WL	4.72	3.50	3.34	3.44	4.50	3.36	4.55
TL	6.71	9.72	4.21	4.67	6.19	4.25	6.20
N_{type}	6413	2688	2258	2610	5041	2295	5124
$N_{tok.}$	28.6k	38.6k	40.5k	39.3k	30.0k	40.2k	29.7k

Table 6: Complete results on the 3K Japhug text with various models (20K iterations, 200 supervision sentences). SP stands for SentencePiece, Morf for Morfessor.

ref.	word			morpheme		
sup.	/	word	mor.	/	word	mor.
BP	61.10	76.62	70.30	87.59	93.30	93.16
BR	90.20	81.11	83.03	75.02	57.31	63.84
BF	72.85	78.80	76.14	80.82	71.00	75.76
WP	38.98	54.44	47.49	58.91	50.06	54.29
WR	55.18	57.23	55.00	51.12	32.25	38.54
WF	45.69	55.80	50.97	54.74	39.23	45.08
LP	39.95	49.93	43.51	45.53	25.85	36.55
LR	13.38	37.35	26.32	37.64	47.71	54.56
LF	20.05	42.73	32.80	41.21	33.53	43.77
WL	3.34	4.50	4.09	3.34	4.50	4.09
TL	4.21	6.19	5.43	4.21	6.19	5.43
N_{type}	2258	5041	4077	2258	5041	4077
$N_{tok.}$	40.5k	30.0k	33.1k	40.5k	30.0k	33.1k

Table 7: Complete results on the 3K Japhug text with the word or morpheme segmented reference (ref.), dictionary from 200 supervision sentences (sup.)

segmented, affecting the second word, 'amipasa'. In the supervised model with `d.mix+2gram`, the word is correctly segmented as 'obengi', and the second word is also correct, although not in the supervision dictionary.

Figure 5 presents two of the 200 sentences used for supervision in Mboshi. This means that all the words in the example are in the supervision dictionary, which can explain why words such as 'owoi', 'atyeeli', or 'lekonyi' are correctly segmented in the weakly supervised setting. Yet, some errors remain (e.g. 'adimo' instead of 'adi mo') mainly because of the cooccurrence effect.

reference	atyeeli adi mo lekonyi
unsupervised	at yee li adi mole konyi
supervised	atyeeli adimo lekonyi
reference	nω owoi dzue la baa
unsupervised	nω o wo i dzuela baa
supervised	nω owoi dzue la baa

Figure 5: Examples of Mboshi sentences used for supervision (here, `d.mix+2gram`): '*Termite workers are on the dead wood*' and '*Did you listen to their voices?*'

A.3 Computing environment

Our experiments have been carried out on an Intel® Xeon® Processor E5-2643 v3 (6 cores and 12 threads). With this processor, the baseline `dpseg` model on the 3K Japhug corpus takes around 10 hours for 20,000 iterations of Gibbs sampling.

All results in this paper have been obtained with a random seed of 42. The remaining parameters are presented in Section 4.2.