Controlling Utterance Length in NMT-based Word Segmentation with Attention

Pierre Godard[1,2], Laurent Besacier[2], François Yvon[1]

(1) LIMSI, CNRS, Université Paris-Saclay, F 91 405 Orsay
(2) LIG, CNRS et Université Grenoble Alpes, F 38 000 Grenoble

`pierre.godard@limsi.fr, laurent.besacier@uga.fr, francois.yvon@limsi.fr`

Abstract

One of the basic tasks of computational language documentation (CLD) is to identify word boundaries in an unsegmented phonemic stream. While several unsupervised monolingual word segmentation algorithms exist in the literature, they are challenged in real-world CLD settings by the small amount of available data. A possible remedy is to take advantage of glosses or translation in a foreign, well-resourced, language, which often exist for such data. In this paper, we explore and compare ways to exploit neural machine translation models to perform unsupervised boundary detection with bilingual information, notably introducing a new loss function for jointly learning alignment and segmentation. We experiment with an actual under-resourced language, Mboshi, and show that these techniques can effectively control the output segmentation length.

1. Introduction

All over the world, languages are disappearing at an unprecedented rate, fostering the need for specific tools aimed to aid field linguists to collect, transcribe, analyze, and annotate endangered language data (e.g. [1, 2]). A remarkable effort in this direction has improved the data collection procedures and tools [3, 4], enabling to collect corpora for an increasing number of endangered languages (e.g. [5]).

One of the basic tasks of computational language documentation (CLD) is to identify word or morpheme boundaries in an unsegmented phonemic or orthographic stream. Several unsupervised monolingual word segmentation algorithms exist in the literature, based, for instance, on information-theoretic [6, 7] or nonparametric Bayesian techniques [8, 9]. These techniques are, however, challenged in real-world settings by the small amount of available data.

A possible remedy is to take advantage of glosses or translations in a foreign, well-resourced language (WL), which often exist for such data, hoping that the bilingual context will provide additional cues to guide the segmentation algorithm. Such techniques have already been explored, for instance, in [10, 11] in the context of improving statistical alignment and translation models; and in [12, 13, 14] using Attentional Neural Machine Translation (NMT) models. In these latter studies, word segmentation is obtained by post-processing attention matrices, taking attention information as a noisy proxy to word alignment [15].[1]

In this paper, we explore ways to exploit neural machine translation models to perform unsupervised boundary detection with bilingual information. Our main contribution is a new loss function for jointly learning alignment and segmentation in neural translation models, allowing us to better control the length of utterances. Our experiments with an actual under-resourced language (UL), Mboshi [18], show that this technique outperforms our bilingual segmentation baseline.

2. Recurrent architectures in NMT

In this section, we briefly review the main concepts of recurrent architectures for machine translation introduced in [19, 20, 21].[2] In our setting, the source and target sentences are always observed and we are mostly interested in the attention mechanism that is used to induce word segmentation.

2.1. RNN encoder-decoder

Sequence-to-sequence models transform a variable-length *source* sequence into a variable-length *target* output sequence. In our context, the source sequence is a sequence of words $w_1, \ldots, w_J$ and the target sequence is an unsegmented sequence of phonemes or characters $\omega_1, \ldots, \omega_I$. In the RNN encoder-decoder architecture, an *encoder* consisting of a RNN reads a sequence of word embeddings $e(w_1), \ldots, e(w_J)$ representing the source and produces a dense representation c of this sentence in a low-dimensional vector space. Vector c is then fed to an RNN *decoder* producing the output translation $\omega_1, \ldots, \omega_I$ sequentially.

At each step of the input sequence, the encoder hidden states h_j are computed as:

$$h_j = \phi(e(w_j), h_{j-1}). \tag{1}$$

In most cases, ϕ corresponds to a long short-term memory (LSTM) [25] unit or a gated recurrent unit (GRU) [26], and h_J is used as the fixed-length context vector c initializing the RNN decoder.

On the target side, the decoder predicts each word ω_i, given the context vector c (in the simplest case, h_J, the last

---

[1] This assumption is further discussed in [16] and [17].
[2] The Transformer model of [22] arguably yields better translation performance, but the underlying soft alignments provided by the multi-head, multi-layered attention mechanism are harder to exploit [23, 24].

hidden state of the encoder) and the previously predicted words, using the probability distribution over the output vocabulary V_T:

$$\begin{cases} P(\omega \mid \omega_1, \ldots, \omega_{i-1}, c) = g(\omega_{i-1}, s_i, c) \\ \omega_i = \operatorname{argmax}_{\omega_k} P(\omega = \omega_k \mid \omega_1, \ldots, \omega_{i-1}, c), \end{cases} \quad (2)$$

where s_i is the hidden state of the decoder RNN and g is a nonlinear function (e.g. a multi-layer perceptron with a softmax layer) computed by the output layer of the decoder. The hidden state s_i is then updated according to:

$$s_i = f(s_{i-1}, e(\omega_{i-1}), c), \quad (3)$$

where f again corresponds to the function computed by an LSTM or GRU cell.

The encoder and the decoder are trained jointly to maximize the likelihood of the translation $\mathbf{\Omega} = \Omega_1, \ldots, \Omega_I$ given the source sentence $\mathbf{w} = w_1, \ldots, w_J$. As reference target words are available during training, Ω_i (and the corresponding embedding) can be used instead of ω_i in Equations (2) and (3), a technique known as *teacher forcing* [27].[3]

2.2. The attention mechanism

Encoding a variable-length source sentence in a fixed-length vector can lead to poor translation results with long sentences [20]. To address this problem, [21] introduces an attention mechanism which provides a flexible source context to better inform the decoder's decisions. This means that the fixed context vector c in Equations (2) and (3) is replaced with a position-dependent context c_i, defined as:

$$c_i = \sum_{j=1}^{J} \alpha_{ij} h_j, \quad (4)$$

where weights α_{ij} are computed by an *attention model* made of a multi-layer perceptron (MLP) followed by a softmax layer. Denoting a the function computed by the MLP, then

$$\begin{cases} e_{ij} & = a(s_{i-1}, h_j) \\ \alpha_{ij} & = \frac{\exp(e_{ij})}{\sum_{k=1}^{J} \exp(e_{ik})}, \end{cases} \quad (5)$$

where e_{ij} is known as the *energy* associated to α_{ij}. Lines in the attention matrix $A = (\alpha_{ij})$ sum to 1, and weights α_{ij} can be interpreted as the probability that target word ω_i is aligned to source word w_j. [21] qualitatively investigated such soft alignments and concluded that their model can correctly align target words to relevant source words (see also [28, 29]). Our segmentation method (Section 3) relies on the assumption that the same holds when aligning characters or phonemes on the target side to source words.

3. Attention-based word segmentation

Recall that our goal is to discover words in an unsegmented stream of target characters (or phonemes) in the under-resourced language. In this section, we first describe a baseline method inspired by the "align to segment" of [13, 14]. We then propose two extensions providing the model with a signal relevant to the segmentation process, so as to move towards a joint learning of segmentation and alignment.

3.1. Align to segment

An attention matrix $A = (\alpha_{ij})$ can be interpreted as a soft alignment matrix between target and source units, where each cell α_{ij} corresponds to the probability for target symbols ω_i (here, a phone) to be aligned to the source word w_j (cf. Equation (5)). In our context, where words need to be discovered on the target side, we follow [13, 14] and perform word segmentation as follows:

1. train an attentional RNN encoder-decoder model with attention using teacher forcing (see Section 2);

2. force-decode the entire corpus and extract one attention matrix for each sentence pair.

3. identify boundaries in the target sequences. For each target unit ω_i of the UL, we identify the source word w_{a_i} to which it is most likely aligned : $\forall i, a_i = \operatorname{argmax}_j \alpha_{ij}$. Given these alignment links, a word segmentation is computed by introducing a word boundary in the target whenever two adjacent units are not aligned with the same source word ($a_i \neq a_{i+1}$).

Considering a (simulated) low-resource setting, and building on [15]'s work, [12] propose to smooth attentional alignments, either by post-processing attention matrices, or by flattening the softmax function in the attention model (see Equation (5)) with a temperature parameter T.[4] This makes sense as the authors examine attentional alignments obtained while training from UL phonemes to WL words. But when translating from WL words to UL characters, this seems less useful: smoothing will encourage a character to align to many words.[5] This technique is further explored by [30], who make the temperature parameter trainable and specific to each decoding step, so that the model can learn how to control the softness or sharpness of attention distributions, depending on the current word being decoded.

3.2. Towards joint alignment and segmentation

One limitation in the approach described above lies in the absence of signal relative to segmentation during RNN training. Attempting to move towards a joint learning of alignment and segmentation, we propose here two extensions

---

[3] As discussed below, teacher forcing will also be used at "test" time in our scenario, since we are not training these models for a translation task, but a word segmentation task.

[4] In this case, $\alpha_{ij} = \frac{\exp(e_{ij}/T)}{\sum_{k=1}^{J} \exp(e_{ik}/T)}$.

[5] A temperature below 1 would conversely sharpen the alignment distribution. We did not observe significant changes in segmentation performance varying the temperature parameter.

aimed at introducing constraints derived from our segmentation heuristic in the training process.

3.2.1. Word-length bias

Our first extension relies on the assumption that the length of aligned source and target words should correlate. Being in a relationship of mutual translation, aligned words are expected to have comparable frequencies and meaning, hence comparable lengths.[6] This means that the longer a source word is, the more target units should be aligned to it. We implement this idea in the attention mechanism as a word-length bias, changing the computation of the context vector from Equation (4) to:

$$c_i = \sum_j \psi(|w_j|)\, \alpha_{ij}\, h_j \qquad (6)$$

where ψ is a monotonically increasing function of the length $|w_j|$ of word w_j. This will encourage target units to attend more to longer source words. In practice, we choose ψ to be the identity function and renormalize so as to ensure that lines still sum to 1 in the attention matrices. The context vectors c_i are now computed with attention weights $\tilde{\alpha}_{ij}$ as:

$$\begin{cases} \tilde{\alpha}_{ij} &= \frac{|w_j|}{\sum_j |w_j| \alpha_{ij}}\, \alpha_{ij} \\ c_i &= \sum_j \tilde{\alpha}_{ij}\, h_j . \end{cases} \qquad (7)$$

We finally derive the target segmentation from the attention matrix $A = (\tilde{\alpha}_{ij})$, following the method of Section 3.1.

### 3.2.2. Introducing an auxiliary loss function

Another way to inject segmentation awareness inside our training procedure is to control the number of target words that will be produced during post-processing. The intuition here is that notwithstanding typological discrepancies, the target segmentation should yield a number of target words that is close to the length of the source.[7]

To this end, we complement the main loss function with an additional term $\mathcal{L}_{\text{AUX}}$ defined as:

$$\mathcal{L}_{\text{AUX}}(\boldsymbol{\Omega}\,|\,\mathbf{w}) = \left| I - J - \sum_{i=1}^{I-1} \alpha_{i,*}^{\top} \alpha_{i+1,*} \right| \qquad (8)$$

The rationale behind this additional term is as follows: recall that a boundary is then inserted on the target side whenever two consecutive units are not aligned to the same source word. The dot product between consecutive lines in the attention matrix will be close to 1 if consecutive target units are aligned to the same source word, and closer to 0 if they are not. The summation thus quantifies the number of target

units that will *not* be followed by a word boundary after segmentation, and $I - \sum_{i=1}^{I-1} \alpha_{i,*}^{\top} \alpha_{i+1,*}$ measures the number of word boundaries that are produced on the target side. Minimizing this auxiliary term should guide the model towards learning attention matrices resulting in target segmentations that have the same number of words on the source and target sides.
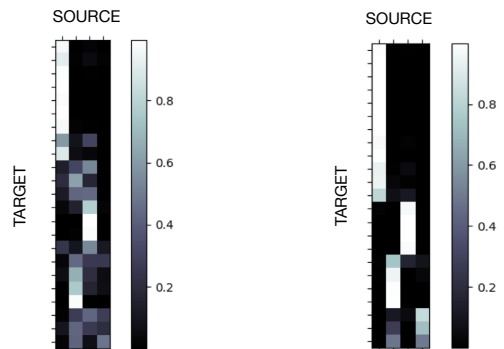
(a) without auxiliary loss (b) with auxiliary loss

Figure 1: Effect of the auxiliary loss ($\mathcal{L}_{\text{NLL}}$) on an example attention matrix for a sentence pair. Lines are indexed by target characters (or phonemes) and columns, by source words; lighter squares correspond to higher attention weights α_{ij}.

Figure 1 illustrates the effect of our auxiliary loss on an example. Without auxiliary loss, the segmentation will yield, in this case, 8 target segments (Figure 1a), while the attention learnt with auxiliary loss will yield 5 target segments (Figure 1b); source sentence, on the other hand, has 4 tokens.[8]

4. Experiments and discussion

In this section, we describe implementation details for our baseline segmentation system and for the extensions proposed in Section 3.2, before presenting data and results.

4.1. Implementation details

Our baseline system is our own reimplementation of Bahdanau's encoder-decoder with attention in PyTorch [32].[9] The last version of our code, which handles mini-batches efficiently, heavily borrows from Joost Basting's code.[10] Source sentences include an end-of-sentence (EOS) symbol (corresponding to w_J in our notation) and target sentences include both a beginning-of-sentence (BOS) and an EOS symbol. Padding of source and target sentences in mini-batches is required, as well as masking in the attention matrices and during loss computation. Our architecture follows [21] very closely with some minor changes.

---

[6]Zipf's "Law of Abbreviation", a language universal, states that frequent words tend to be short. Additionally, experimental psychology also correlates word length and conceptual complexity [31].

[7]Arguably, this constraint should better be enforced at the level of morphemes, instead of words.

[8]We count here the end-of-sentence token corresponding to the last column in the attention matrices.

[9]https://pytorch.org/. We use version 0.4.1.

[10]https://github.com/bastings/annotated_encoder_decoder.

Encoder We use a single-layer bidirectional RNN [33] with GRU cells: these have been shown to perform similarly to LSTM-based RNNs [34], while computationally more efficient. We use 64-dimensional hidden states for the forward and backward RNNs, and for the embeddings, similarly to [13, 14]. In Equation (1), h_j corresponds to the concatenation of the forward and backward states for each step j of the source sequence.

Attention The alignment MLP model computes function a from Equation (5) as $a(s_{i-1}, h_j) = v_a^\top \tanh(W_a s_{i-1} + U_a h_j)$ – see Appendix A.1.2 in [21] – where v_a, W_a, and U_a are weight matrices. For the computation of weights $\widetilde{\alpha_{ij}}$ in the word-length bias extension (Equation (7)), we arbitrarily attribute a length of 1 to the EOS symbol on the source side.

Decoder The decoder is initialized using the last backward state of the encoder and a non-linear function ($\tanh$) for state s_0. We use a single-layer GRU RNN; hidden states and output embeddings are 64-dimensional. In preliminary experiments, and as in [35], we observed better segmentations adopting a "generate first" approach during decoding, where we first generate the current target word, then update the current RNN state. Equations (2) and (3) are accordingly modified into:

$$\begin{cases} P(\omega_i \mid \omega_1, \dots, \omega_{i-1}, c_i) = g(\omega_{i-1}, s_{i-1}, c_i) \\ s_i = f(s_{i-1}, \omega_i, c_i). \end{cases}$$

During training and forced decoding, the hidden state s_i is thus updated using ground-truth embeddings $e(\Omega_i)$. Ω_0 is the BOS symbol. Our implementation of the output layer (g) consists of a MLP and a softmax.

Training We train for 800 epochs on the whole corpus with Adam (the learning rate is 0.001). Parameters are updated after each mini-batch of 64 sentence pairs.[11] A dropout layer [36] is applied to both source and target embedding layers, with a rate of 0.5.[12] The weights in all linear layers are initialized with Glorot's normalized method (Equation (16) in [37]) and bias vectors are initialized to 0. Embeddings are initialized with the normal distribution $\mathcal{N}(0, 0.1)$.[13] Except for the bridge between the encoder and the decoder, the initialization of RNN weights is kept to PyTorch defaults. During training, we minimize the NLL loss $\mathcal{L}_{\mathrm{NLL}}$ (see Section 2.1), adding optionally the auxiliary loss $\mathcal{L}_{\mathrm{AUX}}$ (Section 3.2.2). When the auxiliary loss term is used, we schedule it to be integrated progressively so as to

avoid degenerate solutions[14] with coefficient $\lambda_{\mathrm{AUX}}(k)$ at epoch k defined by:

$$\lambda_{\mathrm{AUX}}(k) = \frac{\max(k - W)}{K} \tag{9}$$

where K is the total number of epochs and W a *wait* parameter. The complete loss at epoch k is thus $\mathcal{L}_{\mathrm{NLL}} + \lambda_{\mathrm{AUX}} \cdot \mathcal{L}_{\mathrm{AUX}}$. After trying values ranging from 100 to 700, we set W to 200. We approximate the absolute value in Equation (8) by $|x| \triangleq \sqrt{x^2 + 0.001}$, in order to make the auxiliary loss function differentiable.

4.2. Data and evaluation

Our experiments are performed on an actual endangered language, Mboshi (Bantu C25), a language spoken in Congo-Brazzaville, using the bilingual French-Mboshi 5K corpus of [18]. On the Mboshi side, we consider alphabetic representation with no tonal information. On the French side, we simply consider the default segmentation into words.[15]

We denote the baseline segmentation system as BASE, the word-length bias extension as BIAS, and the auxiliary loss extensions as AUX. We also report results for a variant of AUX (AUX+RATIO), in which the auxiliary loss is computed with a factor corresponding to the true length ratio $r_{\mathrm{MB/FR}}$ between Mboshi and French averaged over the first 100 sentences[16] of the corpus. In this variant, the auxiliary loss is computed as $|I - r_{\mathrm{MB/FR}} \cdot J - \sum_{i=1}^{I-1} \alpha_{i,*}^\top \alpha_{i+1,*}|$.

We report segmentation performance using precision, recall, and F-measure on boundaries (BP, BR, BF), and tokens (WP, WR, WF). We also report the exact-match (X) metric which computes the proportion of correctly segmented utterances.[17] Our main results are in Figure 2, where we report averaged scores over 10 runs. As a comparison with another bilingual method inspired by the "align to segment" approach, we also include the results obtained using the statistical models of [10], denoted `Pisa`, in Table 1.

BP	BR	BF	WP	WR	WF	X
46.18	18.31	26.22	17.73	8.82	11.78	0.97

Table 1: Equivalent segmentation results with `Pisa` [10].

---

[11]Mini-batches are created anew through shuffling and length-sorting at each epoch.

[12]We also tried to add a dropout layer after the encoder and decoder RNNs (only for the "output" state, not the state values used inside the recursion) but this harmed our segmentation results.

[13]This seemed to slightly improve segmentation results when compared to Glorot's normalized method.

[14]When the NLL loss has not "shaped" yet the attention matrices into soft alignments, the auxiliary loss can lead to trivial optimization solutions, in which a single column in the attention matrices has a certain number of weights set to 1 (to reach the proper value in the sum term from Equation (8)), while all other weights in the matrices are zeroed. The model is subsequently unable to escape this solution.

[15]Several other units were also considered (lemmas, morphs, part-of-speech tags) but they did not lead to better performance (results not reported here). Investigation of subword units such as byte pair encodings (BPE) is left for future work.

[16]This is a plausible supervision in the CLD scenario, and enables to relax assumption that number of words should be the same on target and source.

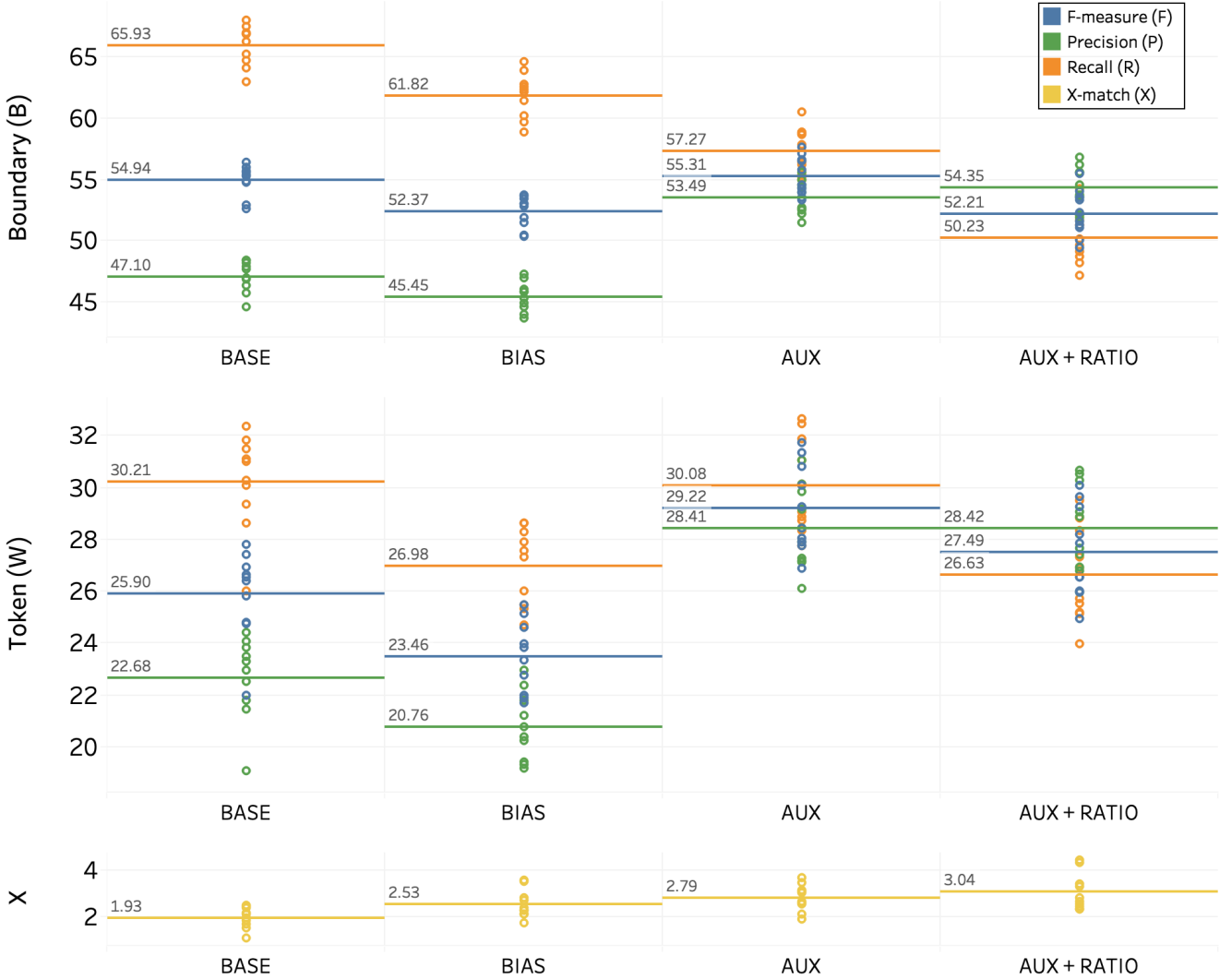[17]The exact-match metric includes single-word utterances.

Figure 2: Boundary and token metrics (F-measure, precision, recall), and sentence exact-match (X) with methods BASE, BIAS, AUX, and AUX+RATIO, on the Mboshi 5K corpus. Horizontal colored lines correspond to values averaged over the 10 runs.

4.3. Discussion

A first observation is that our baseline method BASE improves vastly over `Pisa`'s results (by a margin of about 30% on boundary F-measure, BF).

4.3.1. Effects of the word-length bias

The integration of a word-bias in the attention mechanism seems detrimental to segmentation performance, and results obtained with BIAS are lower than those obtained with BASE, except for the sentence exact-match metric (X). To assess whether the introduction of word-length bias actually encourages target units to "attend more" to longer source word in BIAS, we compute the correlation between the length of source word and the quantity of attention these words receive (for each source position, we sum attention column-wise: $\sum_i \tilde{\alpha}_{ij}$). Results for all segmentation methods are in

Table 2. BIAS increases the correlation between word lengths and attention, but this correlation being already high for all methods (BASE, or AUX and AUX+RATIO), our attempt to increase it proves here detrimental to segmentation.

BASE	BIAS	AUX	AUX+RATIO
0.681	**0.729**	0.665	0.662

Table 2: Correlation (avg. over 10 runs) between word length and attention (p-value for Pearson coefficient is 0 for each run) for methods BASE, BIAS, AUX, and AUX+RATIO.

4.3.2. Effects of the auxiliary loss

For boundary F-measures (BF) in Figure 2, AUX performs similarly to BASE, but with a much higher precision, and de-

method	#tokens	#types	avg. tok. len	avg. sent. len
BASE	40.7K	8.2K	3.14	7.94
BIAS	39.7K	8.9K	3.22	7.75
AUX	32.3K	9.1K	3.95	6.31
AUX+RATIO	28.6K	9.6K	4.47	5.58
ground-truth	*30.6K*	*5.3K*	*4.19*	*5.96*

Table 3: Statistics on segmentations produced by methods BASE, BIAS, AUX, and AUX+RATIO, on the Mboshi 5K corpus: number of tokens, types, average token length (in characters), average sentence lengths (in tokens), averaged over 10 runs.

graded recall, indicating that the new method does not over-segment as much as BASE. More insight can be gained from various statistics on the automatically segmented data presented in Table 3. The average token and sentence lengths for AUX are closer to their ground-truth values (resp. 4.19 characters and 5.96 words). The global number of tokens produced is also brought closer to its reference. On token metrics, a similar effect is observed, but the trade-off between a lower recall and an increased precision is more favorable and yields more than 3 points in F-measure. These results are encouraging for documentation purposes, where precision is arguably a more valuable metric than recall in a semi-supervised segmentation scenario.

They, however, rely on a crude heuristic that the source and target sides (here French and Mboshi) should have the same number of units, which are only valid for typologically related languages and not very accurate for our dataset.

As Mboshi is more agglutinative than French (5.96 words per sentence on average in the Mboshi 5K, vs. 8.22 for French), we also consider the lightly supervised setting where the true length ratio is provided. This again turns out to be detrimental to performance, except for the boundary precision (BP) and the sentence exact-match (X). Note also that precision becomes stronger than recall for both boundary and token metrics, indicating under-segmentation. This is confirmed by an average token length that exceeds the ground-truth (and an average sentence length below the true value, see Table 3).

Here again, our control of the target length proves effective: compared to BASE, the auxiliary loss has the effect to decrease the average sentence length and move it closer to its observed value (5.96), yielding an increased precision, an effect that is amplified with AUX+RATIO. By tuning this ratio, it is expected that we could even get slightly better results.

5. Related work

The attention mechanism introduced by [21] has been further explored by many researchers. [38], for instance, compare a *global* to a *local* approach for attention, and examine several architectures to compute alignment weights α_{ij}. [39] addi-

tionally propose a recurrent version of the attention mechanism, where a "dynamic memory" keeps track of the attention received by each source word, and demonstrate better translation results. A more general formulation of the attention mechanism can, lastly, be found in [40], where structural dependencies between source units can be modeled.

With the goal of improving alignment quality, [41] computes a distance between attentions and word alignments learnt with the reparameterization of IBM Model 2 from [42]; this distance is then added to the cost function during training. To improve alignments also, [15] introduce several refinements to the attention mechanism, in the form of structural biases common in word-based alignment models. In this work, the attention model is enriched with features able to control positional bias, fertility, or symmetry in the alignments, which leads to better translations for some language pairs, under low-resource conditions. More work seeking to improve alignment and translation quality can be found in [43, 44, 45, 46, 47, 48].

Another important line of reseach related to work studies the relationship between segmentation and alignment quality: it is recognized that sub-lexical units such as BPE [49] help solve the unknown word problem; other notable works around these lines include [50] and [51].

CLD has also attracted a growing interest in recent years. Most recent work includes speech-to-text translation [52, 53], speech transcription using bilingual supervision [54], both speech transcription and translation [55], or automatic phonemic transcription of tonal languages [56].

6. Conclusion

In this paper, we explored neural segmentation methods extending the "align to segment" approach, and proposed extensions to move towards joint segmentation and alignment. This involved the introduction of a word-length bias in the attention mechanism and the design of an auxiliary loss. The latter approach yielded improvements over the baseline on all accounts, in particular for the precision metric.

Our results, however, lag behind the best monolingual performance for this dataset (see e.g. [57]). This might be due to the difficulty of computing valid alignments between phonemes and words in very limited data conditions, which remains very challenging, as also demonstrated by the results of Pisa. However, unlike monolingual methods, bilingual methods generate word alignments and their real benefit should be assessed with alignment based metrics. This is left for future work, as reference word alignments are not yet available for our data.

Other extensions of this work will focus on ways to mitigate data sparsity with weak supervision information, either by using lists of frequent words or the presence of certain word boundaries on the target side or by using more sophisticated attention models in the spirit of [15] or [40].

7. References

[1] A. Anastasopoulos, S. Bansal, D. Chiang, S. Goldwater, and A. Lopez, "Spoken term discovery for language documentation using translations," in *Proceedings of the Workshop on Speech-Centric Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 53–58. [Online]. Available: http://aclweb.org/anthology/W17-4607

[2] O. Adams, T. Cohn, G. Neubig, and A. Michaud, "Phonemic transcription of low-resource tonal languages," in *Proceedings of the Australasian Language Technology Association Workshop 2017*, 2017, pp. 53–60. [Online]. Available: http://aclweb.org/anthology/U17-1006

[3] S. Bird, F. R. Hanke, O. Adams, and H. Lee, "Aikuma: A mobile app for collaborative language documentation," in *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, Baltimore, MA, 2014, pp. 1—-5.

[4] D. Blachon, E. Gauthier, L. Besacier, G.-N. Kouarata, M. Adda-Decker, and A. Rialland, "Parallel speech collection for under-resourced language studies using the LIG-Aikuma mobile device app," *Procedia Computer Science*, vol. 81, pp. 61–66, 2016.

[5] G. Adda, S. Stüker, M. Adda-Decker, O. Ambouroue, L. Besacier, D. Blachon, H. Bonneau-Maynard, P. Godard, F. Hamlaoui, D. Idiatov, G.-N. Kouarata, L. Lamel, E.-M. Makasso, A. Rialland, M. Van de Velde, F. Yvon, and S. Zerbian, "Breaking the unwritten language barrier: The Bulb project," in *Proceedings of SLTU (Spoken Language Technologies for Under-Resourced Languages)*, Yogyakarta, Indonesia, 2016.

[6] M. Creutz, "Unsupervised segmentation of words using prior distributions of morph length and frequency," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, July 2003, pp. 280–287. [Online]. Available: http://www.aclweb.org/anthology/P03-1036

[7] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Trans. Speech Lang. Process.*, vol. 4, no. 1, pp. 3:1–3:34, Feb. 2007. [Online]. Available: http://doi.acm.org/10.1145/1187415.1187418

[8] S. Goldwater, "Nonparametric Bayesian models of lexical acquisition," Ph.D. dissertation, Brown University, 2006.

[9] M. Johnson, "Unsupervised Word Segmentation for Sesotho Using Adaptor Grammars," in *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*. Columbus, Ohio: Association for Computational Linguistics, 2008, pp. 20–27.

[10] F. Stahlberg, T. Schlippe, S. Vogel, and T. Schultz, "Word segmentation through cross-lingual word-to-phoneme alignment," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 85–90.

[11] G. Neubig, T. Watanabe, S. Mori, and T. Kawahara, "Machine translation without words through substring alignment," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 165–174. [Online]. Available: http://www.aclweb.org/anthology/P12-1018

[12] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 949–959.

[13] M. Zanon Boito, A. Bérard, A. Villavicencio, and L. Besacier, "Unwritten Languages Demand Attention Too! Word Discovery with Encoder-Decoder Models," in *Automatic Speech Recognition and Understanding (ASRU), 2017 IEEE Workshop On*. IEEE, 2017.

[14] P. Godard, M. Zanon Boito, L. Ondel, A. Bérard, F. Yvon, A. Villavicencio, and L. Besacier, "Unsupervised Word Segmentation from Speech with Attention," in *Proceedings of Interspeech*, Hyderabad, India, 2018.

[15] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari, "Incorporating structural alignment biases into an attentional neural translation model," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 876–885.

[16] S. Jain and B. C. Wallace, "Attention is not Explanation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3543–3556.

[17] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," 2019.

[18] P. Godard, G. Adda, M. Adda-Decker, J. Benjumea, L. Besacier, J. Cooper-Leavitt, G. Kouarata, L. Lamel,

H. Maynard, M. Müller, A. Rialland, S. Stüker, F. Yvon, and M. Z. Boito, "A Very Low Resource Language Speech Corpus for Computational Language Documentation Experiments," in *Proceedings of the Language Resource and Evaluation Conference*, Miyazaki, Japan, 2018.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[20] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the first International Conference on Learning Representations*, ser. ICLR 2015, San Diego, CA, 2015.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, June 2017.

[23] T. Alkhouli, G. Bretschner, and H. Ney, "On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 177–185.

[24] M. Zanon Boito, A. Villavicencio, and L. Besacier, "Empirical Evaluation of Sequence-to-Sequence Models for Word Discovery in Low-resource Settings," in *Interspeech 2019*, Graz, Austria, Sept. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02193867

[25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[26] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[27] R. J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, June 1989.

[28] P. Koehn and R. Knowles, "Six Challenges for Neural Machine Translation," in *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, Aug. 2017, pp. 28–39.

[29] H. Ghader and C. Monz, "What does Attention in Neural Machine Translation Pay Attention to?" *arXiv:1710.03348 [cs]*, Oct. 2017.

[30] J. Lin, X. Sun, X. Ren, M. Li, and Q. Su, "Learning when to concentrate or divert attention: Self-adaptive attention temperature for neural machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP 2018. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2985–2990.

[31] M. L. Lewis and M. C. Frank, "The length of words reflects their conceptual complexity," *Cognition*, vol. 153, pp. 182–195, Aug. 2016.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.

[33] M. Schuster and K. Paliwal, "Bidirectional Recurrent Neural Networks," *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[35] J.-T. Peter, A. Nix, and H. Ney, "Generating Alignments Using Target Foresight in Attention-Based Neural Machine Translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, no. 1, pp. 27–36, June 2017.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256.

[38] T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.

[39] Z. Yang, Z. Hu, Y. Deng, C. Dyer, and A. Smola, "Neural Machine Translation with Recurrent Attention Modeling," *arXiv preprint arXiv:1607.05108*, 2016.

[40] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured Attention Networks," in *5th International Conference on Learning Representations*, 2017, p. 21.

[41] H. Mi, Z. Wang, and A. Ittycheriah, "Supervised Attentions for Neural Machine Translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2283–2288.

[42] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 644–648.

[43] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling Coverage for Neural Machine Translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 76–85.

[44] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Neural machine translation with supervised attention," in *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, ser. COLING 2016. Osaka, Japan: The COLING 2016 Organizing Committee, 2016, pp. 3093–3102.

[45] B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah, "Temporal Attention Model for Neural Machine Translation," Aug. 2016.

[46] S. Feng, S. Liu, M. Li, and M. Zhou, "Implicit Distortion and Fertility Models for Attention-based Encoder-Decoder NMT Model," Jan. 2016.

[47] T. Alkhouli and H. Ney, "Biasing Attention-Based Recurrent Neural Networks Using External Alignment Information," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 108–117.

[48] S. Kuang, J. Li, A. Branco, W. Luo, and D. Xiong, "Attention focusing for neural machine translation by bridging source and target embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 1767–1776. [Online]. Available: https://www.aclweb.org/anthology/P18-1164

[49] R. Sennrich, B. Haddow, and A. Birch, "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)*, Berlin, Germany, Aug. 2016, pp. 1715–1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[50] T. Kudo, "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)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 66–75.

[51] J. Kreutzer and A. Sokolov, "Learning to segment inputs for NMT favors character-level processing," in *Proceedings of the international workshop on spoken language processing*, ser. IWSLT'18, Bruges, Belgium, 2018. [Online]. Available: http://arxiv.org/abs/1810.01480

[52] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-Resource Speech-to-Text Translation," in *Interspeech 2018*. ISCA, Sept. 2018, pp. 1298–1302.

[53] ——, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 58–68. [Online]. Available: https://www.aclweb.org/anthology/N19-1006

[54] A. Anastasopoulos and D. Chiang, "Leveraging translations for speech transcription in low-resource settings," in *Proc. Interspeech 2018*, 2018, pp. 1279–1283. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-2162

[55] ——, "Tied Multitask Learning for Neural Speech Translation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 82–91.

[56] O. Adams, T. Cohn, G. Neubig, H. Cruz, S. Bird, and A. Michaud, "Evaluation phonemic transcription of low-resource tonal languages for language documentation," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*, 2018. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2018/summaries/490.html

[57] P. Godard, L. Besacier, F. Yvon, M. Adda-Decker, G. Adda, H. Maynard, and A. Rialland, "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*. Brussels, Belgium: Association for Computational Linguistics, October 2018, pp. 32–42.
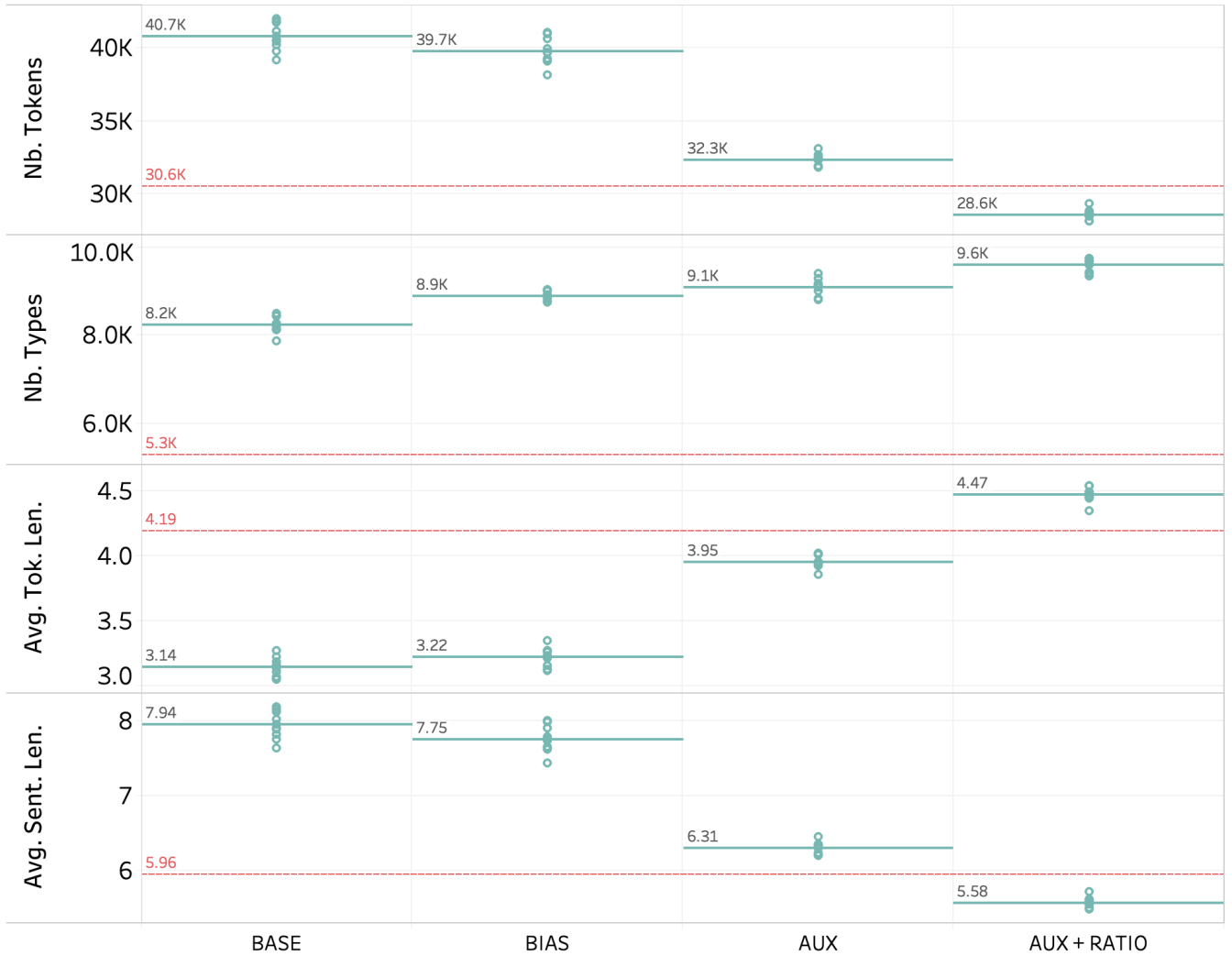
A. Appendix

Figure 3: Statistics on segmentations produced by methods BASE, BIAS, AUX, and AUX+RATIO, on the Mboshi 5K corpus: number of tokens, types, average token length (in characters), average sentence lengths (in tokens). Solid (teal-colored) lines correspond to average values (10 runs). Dashed (red) lines indicate the ground-truth values in the Mboshi 5K corpus.