



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/155716/>

Version: Published Version

Proceedings Paper:

Boito, M.Z., Villavicencio, A. and Besacier, L. (2019) Empirical evaluation of sequence-to-sequence models for word discovery in low-resource settings. In: Kubin, G. and Kačič, Z., (eds.) Interspeech 2019 - Proceedings of the Annual Conference of the International Speech Communication Association. Interspeech 2019, 15-19 Sep 2019, Graz, Austria. Interspeech Proceedings. International Speech Communication Association (ISCA), pp. 2688-2692. ISSN: 2308-457X. EISSN: 1990-9772.

<https://doi.org/10.21437/Interspeech.2019-2029>

© 2019 International Speech Communication Association. Reproduced in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



Empirical Evaluation of Sequence-to-Sequence Models for Word Discovery in Low-resource Settings

Marcelly Zanon Boito¹, Aline Villavicencio^{2,3}, Laurent Besacier¹

¹Laboratoire d'Informatique de Grenoble, Univ. Grenoble Alpes (UGA), France

²School of Computer Science and Electronic Engineering, University of Essex, UK

³Institute of Informatics, Federal University of Rio Grande do Sul, Brazil

marcelly.zanon-boito@univ-grenoble-alpes.fr

Abstract

Since Bahdanau et al. [1] first introduced attention for neural machine translation, most sequence-to-sequence models made use of attention mechanisms [2, 3, 4]. While they produce soft-alignment matrices that could be interpreted as alignment between target and source languages, we lack metrics to quantify their quality, being unclear which approach produces the best alignments. This paper presents an empirical evaluation of 3 of the main sequence-to-sequence models for word discovery from unsegmented phoneme sequences: CNN, RNN and Transformer-based. This task consists in aligning word sequences in a source language with phoneme sequences in a target language, inferring from it word segmentation on the target side [5]. Evaluating word segmentation quality can be seen as an extrinsic evaluation of the soft-alignment matrices produced during training. Our experiments in a low-resource scenario on Mboshi and English languages (both aligned to French) show that RNNs surprisingly outperform CNNs and Transformer for this task. Our results are confirmed by an intrinsic evaluation of alignment quality through the use Average Normalized Entropy (ANE). Lastly, we improve our best word discovery model by using an alignment entropy confidence measure that accumulates ANE over all the occurrences of a given alignment pair in the collection.

Index Terms: sequence-to-sequence models, soft-alignment matrices, word discovery, low-resource languages, computational language documentation

1. Introduction

Sequence-to-Sequence (S2S) models can solve many tasks where source and target sequences have different lengths. For learning to focus on specific parts of the input at decoding time, most of these models are equipped with attention mechanisms [1, 2, 3, 4, 6]. By-products of the attention are soft-alignment probability matrices, that can be interpreted as alignment between target and source. However, we lack metrics to quantify their quality. Moreover, while these models perform very well in a typical use case, it is not clear how they would be affected by low-resource scenarios.

This paper proposes an empirical evaluation of well-known S2S models for a particular S2S modeling task. This task consists of aligning word sequences in a source language with phoneme sequences in a target language, inferring from it word segmentation on the target side [5]. We concentrate on three models: Convolutional Neural Networks (CNN) [2], Recurrent Neural Networks (RNN) [1] and Transformer-based models [3]. While this *word segmentation* task can be used for the extrinsic evaluation of the soft-alignment probability matrices produced

during S2S learning, we also introduce Average Normalized Entropy (ANE), a task-agnostic confidence metric to quantify the quality of the source-to-target alignments obtained. Experiments performed on a low-resource scenario for two languages (Mboshi and English) using equivalently sized corpora aligned to French, are, to our knowledge, the first empirical evaluation of these well-known S2S models for a word segmentation task. We also illustrate how our entropy-based metric can be used in a language documentation scenario, helping a linguist to efficiently discover types, in an unknown language, from an unsegmented sequence of phonemes. This work is thus also a contribution to the emerging computational language documentation domain [7, 8, 9, 10, 11], whose main goal is the creation of automatic approaches able to help the documentation of the many languages soon to be extinct [12].

Lastly, studies focused on comprehensive attention mechanisms for NMT [13, 14, 15] lack evaluation of the resulting alignments, and the exceptions [16] do so for the task of word-to-word alignment in well-resourced languages. Differently, our work is not only an empirical evaluation of NMT models focused on alignment quality, but it also tackles data scarcity of low-resource scenarios.

2. Experimental Settings

2.1. Unsupervised Word Segmentation from Speech

As in language documentation scenarios available corpora usually contain speech in the language to document aligned with translations in a well-resourced language, Godard et al. [5] introduced a pipeline for performing Unsupervised Word Segmentation (UWS) from speech. The system outputs timestamps delimiting stretches of speech, associated with class labels, corresponding to real words in the language. The pipeline consists of first transforming speech into a sequence of phonemes, either through Automatic Unit Discovery (e.g. [17]) or manual transcription. The phoneme sequences, together with their translations, are then fed to an attention-based S2S system that produces soft-alignment probability matrices between target and source languages. The alignment probability distributions between the phonemes and the translation words (as in Figure 1) are used to cluster (segment) together neighbor phonemes whose alignment distribution peaks at the same word. The final speech segmentation is evaluated using the *Zero Resource Challenge*¹ (ZRC) 2017 evaluation suite (track 2).²

¹Available at <http://zerospeech.com/2017>.

²We increment over [5] by removing silence labels before training, and using them for segmentation. This results in slightly better scores.

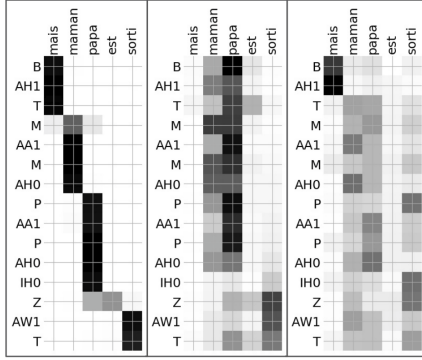


Figure 1: *Soft-alignment probability matrices from the UWS task. ANE values (from left to right) are 0.11, 0.64 and 0.83. The gold segmentation is “BAHIT MAAIMAH0 PAAIPAH0 IH0Z AWIT”, which corresponds to the English sentence “But mama, papa is out”.*

2.2. Parallel Speech Corpora

The parallel speech corpora used in this work are the English-French (EN-FR) [18] and the Mboshi-French (MB-FR) [19] parallel corpora. EN-FR corpus is a 33,192 sentences multilingual extension from *librispeech* [20], with English audio books automatically aligned to French translations. MB-FR is a 5,130 sentences corpus from the language documentation process of Mboshi (Bantu C25), an endangered language spoken in Congo-Brazzaville. Thus, while the former corpus presents larger vocabulary and longer sentences, the latter presents a more tailored environment, with short sentences and simpler vocabulary. In order to provide a fair comparison, as well as to study the impact of corpus size, the EN-FR corpus was also down-sampled to 5K utterances (to the exact same size than the MB-FR corpus). Sub-sampling was conducted preserving the average number of tokens per sentence, shown in Table 1.

2.3. Introducing Average Normalized Entropy (ANE)

In this paper, we focus on studying the soft-alignment probability matrices resulting from the learning of S2S models for the UWS task. To assess the overall quality of these matrices without having gold alignment information, we introduce Average Normalized Entropy (ANE).

Definition: Given the source and target pair (s, t) of lengths $|s|$ and $|t|$ respectively, for every phone t_i , the normalized entropy (NE) is computed considering all possible words in s (Equation 1), where $P(t_i, s_j)$ is the alignment probability between the phone t_i and the word s_j (a cell in the matrix). The ANE for a sentence is then defined by the arithmetic mean over the resulting NE for every phone from the sequence t (Equation 2).

$$NE(t_i, s) = - \sum_{j=1}^{|s|} P(t_i, s_j) \cdot \log_{|s|}(P(t_i, s_j)) \quad (1)$$

$$ANE(t, s) = \frac{\sum_{i=1}^{|t|} NE(t_i, s)}{|t|} \quad (2)$$

From this definition, we can derive ANE for different granularities (sub or supra-sentential) by accumulating its value for the full corpus, for a single type or for a single token. *Corpus ANE* will be used to summarize the overall performance of a S2S model on a specific corpus. *Token ANE* extends ANE to tokens

by averaging NE for all phonemes from a single (discovered) token. *Type ANE* results from averaging the ANE for every token instance of a discovered type. Finally, *Alignment ANE* is the result of averaging the ANE for every discovered (*type, translation word*) alignment pair. Intuition that lower ANEs correspond to better alignments is exemplified in Figure 1.

3. Empirical Comparison of S2S Models

We compare three NMT models (§3.1, §3.2, §3.3) for UWS, focusing on their ability of aligning words (French) with phonemes (English or Mboshi) in medium-low resource settings. The results, an analysis of the impact of data size and quality, and the correlation between intrinsic (ANE) and extrinsic (boundary F-score) metrics are presented in §3.4. The application of ANE for type discovery in low-resource settings is presented in §3.5.

3.1. RNN: Attention-based Encoder-Decoder

The classic RNN encoder-decoder model [1] connects a bidirectional encoder with an unidirectional decoder by the use of an *alignment module*. The RNN encoder learns annotations for every source token, and these are weighted by the alignment module for the generation of every target token. Weights are defined as *context vectors*, since they capture the importance of every source token for the generation of each target token.

Attention mechanism: a context vector for a decoder step t is computed using the set of source annotations H and the last state of the decoder network (translation context). The attention is the result of the weighted sum of the source annotations H (with $H = h_1, \dots, h_A$) and their α probabilities (3) obtained through a feed-forward network *align* (4).

$$c_t = \text{Att}(H, s_{t-1}) = \sum_{i=1}^A \alpha_i^t h_i \quad (3)$$

$$\alpha_i^t = \text{softmax}(\text{align}(h_i, s_t)) \quad (4)$$

3.2. Transformer

Transformer [3] is a fully attentional S2S architecture, which has obtained state-of-the-art results for several NMT shared tasks. It replaces the use of sequential cell units (such as LSTM) by Multi-Head Attention (MHA) operations, which make the architecture considerably faster. Both encoder and decoder networks are stacked layers sets that receive source and target sequences, embedded and concatenated with positional encoding. An encoder layer is made of two sub-layers: a *Self-Attention* MHA and a feed-forward. A decoder layer is made of three sub-layers: a *masked Self-Attention* MHA (no access to subsequent positions); an *Encoder-Decoder* MHA (operation over the encoder stack’s final output and the decoder self-attention output); and a feed-forward sub-layer. Dropout and residual connections are applied between all sub-layers. Final output probabilities are generated by applying a linear projection over the decoder stack’s output, followed by a softmax operation.

Multi-head attention mechanism: attention is seen as a mapping problem: given a pair of key-value vectors and a query vector, the task is the computation of the weighted sum of the given values (output). In this setup, weights are learned by compatibility functions between key-query pairs (of dimension d_k). For a given query (Q), keys (K) and values (V) set, the *Scaled*

Table 1: Statistics of the three source-target data sets.

corpus	#types		#tokens		average(token length)		average(#tokens / sentence)	
	source	target	source	target	source	target	source	target
EN-FR (33k)	21,083	33,135	381,044	467,475	4.37	4.57	11.48	14.08
EN-FR (5k)	8,740	12,226	59,090	72,670	4.38	4.57	11.52	14.17
MB-FR (5k)	6,633	5,162	30,556	42,715	4.18	4.39	5.96	8.33

Dot-Product (SDP) Attention function is computed as:

$$Att(V, K, Q) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (5)$$

In practice, several attentions are computed for a given QKV set. The QKV set is first projected into h different spaces (multiple heads), where the SDP attention is computed in parallel. Resulting values for all heads are then concatenated and once again projected, yielding the layer’s output. (6) and (7) illustrate the process, in which H is the set of h heads ($H = h_1, \dots, h_h$) and f is a linear projection. **Self-Attention** defines the case where query and values come from same source (learning compatibility functions within the same sequence of elements).

$$MultiHead(V, K, Q) = f(Concat(H)) \quad (6)$$

$$h_i = Att(f_i(V), f_i(K), f_i(Q)) \quad (7)$$

3.3. CNN: Pervasive Attention

Different from the previous models, which are based on encoder-decoder structures interfaced by attention mechanisms, this approach relies on a single 2D CNN across both sequences (no separate coding stages) [2]. Using masked convolutions, an auto-regressive model predicts the next output symbol based on a joint representation of both input and partial output sequences. Given a source-target pair (s, t) of lengths $|s|$ and $|t|$ respectively, tokens are first embed in d_s and d_t dimensional spaces via look-up tables. Token embeddings $\{x_1, \dots, x_{|s|}\}$ and $\{y_1, \dots, y_{|t|}\}$ are then concatenated to form a 3D tensor $X \in \mathbb{R}^{|t| \times |s| \times f_0}$, with $f_0 = d_t + d_s$, where:

$$X_{ij} = [y_i \ x_j] \quad (8)$$

Each convolutional layer $l \in \{1, \dots, L\}$ of the model produces a tensor H_l of size $|t| \times |s| \times f_l$, where f_l is the number of output channels for that layer. To compute a distribution over the tokens in the output vocabulary, the second dimension of the tensor is used. This dimension is of variable length (given by the input sequence) and it is collapsed by max or average pooling to obtain the tensor H_L^{pool} of size $|t| \times f_L$. Finally, 1×1 convolution followed by a softmax operation are applied, resulting in the distribution over the target vocabulary for the next output token.

Attention mechanism: joint encoding acts as an attention-like mechanism, since individual source elements are re-encoded as the output is generated. The self-attention approach of [21] is applied. It computes the attention weight tensor α , of size $|t| \times |s|$, from the last activation tensor H_L , to pool the elements of the same tensor along the source dimension, as follows:

$$\alpha = softmax(W_1 \tanh(H_L W_2)) \quad (9)$$

$$H_L^{Att} = \alpha H_L. \quad (10)$$

where $W_1 \in \mathbb{R}^{f_a}$ and $W_2 \in \mathbb{R}^{f_a \times f_L}$ are weight tensors that map the f_L dimensional features in H_L to the attention weights via an f_a dimensional intermediate representation.

3.4. Comparing S2S Architectures

For each S2S architecture, and each of the three corpora, we train five models (runs) with different initialization seeds.³ Before segmenting, we average the produced matrices from the five different runs as in [5]. Evaluation is done in a *bilingual* segmentation condition that corresponds to the real UWS task. In addition, we also perform segmentation in a *monolingual* condition, where a phoneme sequence is segmented with regards to the corresponding word sequence (transcription) in the same language (hence monolingual).⁴ Our networks are optimized for the monolingual task. Across all architectures, we use embeddings of size 64 and batch size of 32 (5K data set), or embeddings of size 128 and batch size of 64 (33K data set). Dropout of 0.5 and early-stopping procedure are applied in all cases. RNN models have only one layer, a bi-directional encoder, and cell size equal to the embedding size, as in [5]. CNN models use the hyper-parameters from [2] with only 3 layers (5K data set), or 6 (33K data set), and kernel size of 3. Transformer models were optimized starting from the original hyper-parameters of [3]. Best results (among 50 setups) were achieved using 2 heads, 3 layers (encoder and decoder), warm-up of 5K steps, and using cross-entropy loss without label-smoothing. Finally, for selecting which head to use for UWS, we experimented using the last layer’s averaged heads, or by selecting the head with minimum corpus ANE. While the results were not significantly different, we kept the ANE selection.

3.4.1. Unsupervised Word Segmentation Results

The word boundary F-scores⁵ for the task of UWS from phoneme sequence (in Mboshi or English) are presented in Table 2, with monolingual results shown for information only (topline). Surprisingly, RNN models outperform the more recent (CNN and Transformer) approaches. One possible explanation is the lower number of parameters (for a 5K setup, in average 700K parameters are trained, while CNN needs an additional 30.79% and Transformer 5.31%). However, for 33K setups, CNNs actually need 30% less parameters than RNNs, but still perform worse. Transformer’s low performance could be due to the use of several heads “distributing” alignment information across different matrices. Nonetheless, we evaluated averaged heads and single-head models, and these resulted in significant decreases in performance. This suggests that this architecture may not need to learn explicit alignment to translate, but instead it could be capturing different kinds of linguistic information, as discussed in the original paper and in its examples [3]. Also, on the decoder side, the behavior of the self-attention mechanism on phoneme units is unclear and under-

³RNN, CNN and Transformer implementations from [22, 2, 23] respectively.

⁴This task can be seen as an automatic extraction of a pronunciation lexicon from parallel words/phonemes sequences.

⁵For CNN and RNN, average standard deviation for the bilingual task is of less than 0.8%. For Transformer, it is almost 4%.

Table 2: Boundary F-scores for the UWS task.

		Bilingual	Monolingual
EN 33K	RNN	77.10	99.80
	CNN	71.30	98.60
	Transformer	52.70	94.90
EN 5K	RNN	70.40	99.30
	CNN	55.90	98.80
	Transformer	52.50	80.90
MB 5K	RNN	74.00	92.50
	CNN	68.20	89.80
	Transformer	66.40	83.50

studied so far. For the encoder, Voita et al. [15] performed after-training encoder head removal based on *head confidence*, showing that after initial training, most heads were not necessary for maintaining translation performance. Hence, we find the Multi-head mechanism interpretation challenging, and maybe not suitable for a direct word segmentation application, such as our method.

As in [24], our best UWS method (RNN) for the bilingual task does not reach the performance level of a strong Bayesian baseline [25] with F-scores of 89.80 (EN33K), 87.93 (EN5K), and 77.00 (MB5K). However, even if we only evaluate word segmentation performance, our neural approaches learn to segment *and* align, whereas this baseline only learns to segment. Section 3.5 will leverage those alignments for a type discovery task useful in language documentation.

The Pearson’s ρ correlation coefficients between ANE and boundary F-scores for all mono and bilingual runs of all corpora ($N = 30$) are -0.98 (RNN), -0.97 (CNN), and -0.66 (Transformer), with p-values smaller than 10^{-5} . These strong negative correlations confirm our hypothesis that lower ANEs correspond to sharper and better alignments.

3.4.2. Impact of Data Size and Quality

EN33K and EN5K results of Table 2 allow us to analyze the impact of data size on the S2S models. For the bilingual task, RNN performance drops by 7% on average, whereas performance drop is bigger for CNN (14-15%). Transformer performs poorly in both cases, and increasing data size from 5K to 33K seems to help only for a trivial task (see *monolingual* results).

The EN5K and MB5K results of Table 2 reflect the impact of language pairs on the S2S models. We know from [26, 27] that English should be easier to segment than Mboshi, and this was confirmed by both *dpseg* and *monolingual* results. However, this trend is not confirmed in the *bilingual* task, where the quality of the (sentence aligned) parallel corpus seems to have more impact (higher boundary F-scores for MB5K than for EN5K for all S2S models). As shown in Table 1, MB-FR corpus has shorter sentences and smaller lexicon diversity, while EN-FR is made of automatically aligned books (noisy alignments), what may explain our experimental results.

3.5. Type Discovery in Low-Resource Settings

We investigate the use of **Alignment ANE** as a confidence measure. From the RNN models, we extract and rank the discovered types by their ANE, and examine if it can be used to separate true words in the discovered vocabulary from the rest. The results for low-resource scenarios (only 5K) in Table 3 suggest that low ANE corresponds to the portion of the discov-

Table 3: Type retrieval results (RNN) using ANE for keeping most confident (type, translation) pairs. For instance, $ANE = 0.4$ means all discovered types have $ANE \leq 0.4$.

ANE	EN 5K			MB 5K		
	P	R	F	P	R	F
0.1	70.97	0.50	1.00	72.13	0.57	1.12
0.2	55.43	3.85	7.20	49.02	2.89	5.46
0.3	44.99	12.51	19.58	38.18	8.14	13.41
0.4	32.81	21.76	26.17	32.63	16.61	22.01
0.5	23.37	28.17	25.54	27.93	23.44	25.49
0.6	18.54	32.41	23.59	24.73	27.61	26.09
0.7	16.23	34.34	22.04	23.00	30.12	26.08
0.8	15.21	35.16	21.23	22.17	30.95	25.84
0.9	15.01	35.31	21.06	22.06	31.05	25.80
all	15.01	35.34	21.07	22.06	31.05	25.80

Table 4: Top 5 low and high ANE ranking for the discovered types (EN5K), with gold transcription and aligned information between parentheses (respectively). “INV” means incorrect type.

	Top Low ANE	Top High ANE
1	SER1 (sir, <i>EOS_token</i>)	AH0 (a, <i>convenable</i>)
2	HHAH1SH (hush, <i>chut</i>)	IH1 (INV, <i>ah</i>)
3	FIH1SHER0 (fisher, <i>fisher</i>)	D (INV, <i>riant</i>)
4	KLER1K (clerk, <i>clerc</i>)	N (INV, <i>obit</i>)
5	KIH1S (kiss, <i>embrasse</i>)	YUW1 (you, <i>diable</i>)

ered vocabulary the network is *confident* about, and these are, in most of the cases, true discovered lexical items (first row, $P \geq 70\%$).⁶ As we keep higher Alignment ANE values, we increase recall but loose precision. This suggests that, in a documentation scenario, ANE could be used as a confidence measure by a linguist to extract a list of types with higher precision, without having to pass through all the discovered vocabulary. Moreover, as exemplified for EN5K in Table 4, we also retrieve aligned information (translation candidates) for the generated lexicon.

4. Conclusions

We presented an empirical evaluation of different architectures (RNN, CNN and Transformer) with respect to their capacity to align word sequences in a source language with phoneme sequences in a target language, inferring from it word segmentation on the target side (UWS task).⁷ Although RNNs have been outperformed by CNN and Transformer-based models for machine translation, for UWS these architectures are still more robust in low-resource scenarios, and present the best segmentation results. We also introduced ANE, an intrinsic measure of alignment quality of S2S models. Accumulating it over the discovered alignments, we showed it can be used as a confidence measure to select true words, increasing Type F-scores.

⁶Type ANE for the retrieval task was also investigated, and results were positive, but slightly worse than the ones from Alignment ANE.

⁷Pointers for corpora, parameters and implementations available at https://gitlab.com/mzboito/attention_study

5. References

- [1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [2] M. Elbayad, L. Besacier, and J. Verbeek, "Pervasive attention: 2d convolutional neural networks for sequence-to-sequence prediction," *arXiv preprint arXiv:1808.03867*, 2018.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [4] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.
- [5] P. Godard, M. Zanon Boito, L. Ondel, A. Berard, F. Yvon, A. Villavicencio, and L. Besacier, "Unsupervised word segmentation from speech with attention," in *Interspeech*, 2018.
- [6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- [7] G. Adda, S. Stüker, M. Adda-Decker, O. Ambourou, L. Besacier, D. Blachon, H. Bonneu-Maynard, P. Godard, F. Hamlaoui, D. Idiatov, G.-N. Kouarata, L. Lamel, E.-M. Makasso, A. Rialland, M. V. de Velde, F. Yvon, and S. Zerbian, "Breaking the unwritten language barrier: The BULB project," *Procedia Computer Science*, vol. 81, pp. 8–14, 2016.
- [8] A. Anastasopoulos and D. Chiang, "A case study on using speech-to-translation alignments for language documentation," *arXiv preprint arXiv:1702.04372*, 2017.
- [9] L. Besacier, B. Zhou, and Y. Gao, "Towards speech translation of non written languages," in *Spoken Language Technology Workshop, 2006. IEEE*. IEEE, 2006, pp. 222–225.
- [10] C. Lignos and C. Yang, "Recession segmentation: simpler on-line word segmentation using limited resources," in *Proceedings of the fourteenth conference on computational natural language learning*. Association for Computational Linguistics, 2010, pp. 88–97.
- [11] C. Bartels, W. Wang, V. Mitra, C. Richey, A. Kathol, D. Vergyri, H. Bratt, and C. Hung, "Toward human-assisted lexical unit discovery without text resources," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 64–70.
- [12] P. K. Austin and J. Sallabank, *The Cambridge handbook of endangered languages*. Cambridge University Press, 2011.
- [13] K. Song, T. Xu, F. Peng, and J. Lu, "Hybrid self-attention network for machine translation," *arXiv preprint arXiv:1811.00253*, 2018.
- [14] J. Li, Z. Tu, B. Yang, M. R. Lyu, and T. Zhang, "Multi-head attention with disagreement regularization," *arXiv preprint arXiv:1810.10183*, 2018.
- [15] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," *arXiv preprint arXiv:1905.09418*, 2019.
- [16] T. Zenkel, J. Wuebker, and J. DeNero, "Adding interpretable attention to neural translation models improves word alignment," *arXiv preprint arXiv:1901.11359*, 2019.
- [17] L. Ondel, L. Burget, and J. Černocký, "Variational inference for acoustic unit discovery," *Procedia Computer Science*, vol. 81, pp. 80–86, 2016.
- [18] A. C. Kocabiyikoglu, L. Besacier, and O. Kraif, "Augmenting librispeech with french translations: A multimodal corpus for direct speech translation evaluation," *arXiv preprint arXiv:1802.03142*, 2018.
- [19] P. Godard, G. Adda, M. Adda-Decker, J. Benjumea, L. Besacier, J. Cooper-Leavitt, G.-N. Kouarata, L. Lamel, H. Maynard, M. Müller *et al.*, "A very low resource language speech corpus for computational language documentation experiments," *arXiv preprint arXiv:1710.03501*, 2017.
- [20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 5206–5210. [Online]. Available: <https://doi.org/10.1109/ICASSP.2015.7178964>
- [21] Z. Lin, M. Feng, C. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," 2017.
- [22] A. Bérard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," *arXiv preprint arXiv:1612.01744*, 2016.
- [23] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.
- [24] P. Godard, "Unsupervised word discovery for computational language documentation," Ph.D. dissertation, Universit Paris-Saclay, 2019.
- [25] S. Goldwater, T. L. Griffiths, and M. Johnson, "A Bayesian framework for word segmentation: Exploring the effects of context," *Cognition*, vol. 112, no. 1, pp. 21–54, 2009.
- [26] A. Fourtassi, B. Börschinger, M. Johnson, and E. Dupoux, "Why is english so easy to segment?" in *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, 2013, pp. 1–10.
- [27] A. Rialland, M. E. Aborobongui, M. Adda-Decker, and L. Lamel, "Dropping of the class-prefix consonant, vowel elision and automatic phonological mining in embosi (bantu c 25)," in *Selected Proceedings of the 44th Annual Conference on African Linguistics*, 2015, pp. 7–10.