UNIVERSITÉ Grenoble Alpes

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

Marcely Zanon Boito[1], Aline Villavicencio[2,3] and Laurent Besacier[1]

[1]LIG, Univ. Grenoble Alpes, **France**
[2]DCS, Univ. of Sheffield, **UK**
[3]INF, UFRGS, **Brazil**

Motivation

Computational **Language Documentation** (CLD)

- 50 to 90% of the currently spoken languages will go extinct before 2100[1]

- Manually documenting all these languages is infeasible

[1] Austin, Sallabank. ***The Cambridge handbook of endangered languages.*** Cambridge University Press, 2011.

Computational Language Documentation (CLD)

- 50 to 90% of the currently spoken languages will go extinct before 2100[1]

- Manually documenting all these languages is infeasible

GOAL: to automatically retrieve information about language structures to speed up language documentation

UNIVERSITÉ Grenoble Alpes

[1]Austin, Sallabank. ***The Cambridge handbook of endangered languages.*** Cambridge University Press, 2011.

Documentation Corpora

- → Small size (difficult to collect)
- → Often lack written form (oral-tradition languages)
- → Parallel information (translations instead of transcriptions)

SPEECH

Translations
to a well-documented language[1]

[1]Adda et al. ***Breaking the unwritten language barrier: The BULB project.*** SLTU 2016.

Documentation Corpora

- → Small size (difficult to collect)
- → Often lack written form (oral-tradition languages)
- → Parallel information (translations instead of transcriptions)

CLD approaches

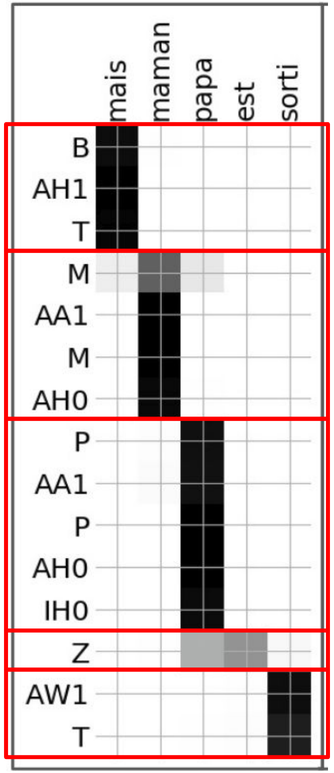
1. Deal with speech
2. Incorporate bilingual (or multilingual) annotations
3. Robust to low-resource

Unsupervised Word Segmentation from Speech with Attention[1]

Low-resource speech word segmentation

We use **Neural Machine Translation** (NMT) model for generating **soft-alignment probability matrices**

Use of the RNN model from Bahdanau et. 2015[2]



[1]Godard, Boito, Ondel, Bérard, Yvon, Villavicencio, Besacier. ***Unsupervised Word Segmentation from Speech with Attention.*** Interspeech 2018.
[2]Bahdanau, Cho, Bengio. **Neural Machine Translation by Jointly Learning to Align and Translate.** ICLR 2015.

But since then...

- SOTA performance with different neural architectures (Transformer[1], convS2S[2], pervasive attention[3], etc) using **different attention mechanisms**

[1]Vaswani et al. ***Attention is all you need.*** NIPS 2017.
[2]Gehring et al. ***Convolutional Sequence to Sequence Learning.*** ICML 2017.
[3]Elbayad, Besacier, Verbeek. ***Pervasive Attention: 2D Convolutional Neural Networks for Sequence-to-Sequence Prediction.*** CoNLL 2018.

But since then...

- SOTA performance with different neural architectures (Transformer[1], convS2S[2], pervasive attention[3], etc) using **different attention mechanisms**

- New discussion about the interpretability of the attention mechanism[4,5,6,7]

[4] Jain, Wallace. ***Attention is not Explanation.*** NAACL 2019.
[5] Wiegreffe, Pinter. ***Attention is not not Explanation.*** EMNLP 2019.
[6] Ghader, Monz. ***What does Attention in Neural Machine Translation Pay Attention to?*** IJCNLP 2017.
[7] Serrano, Smith. **Is Attention Interpretable?** ACL 2019.

Contribution

Empirical evaluation of 3 attention-based seq2seq models

Using different architectures, we want to investigate:

→ Their capacity of generating exploitable alignment

→ Their robustness to data scarcity

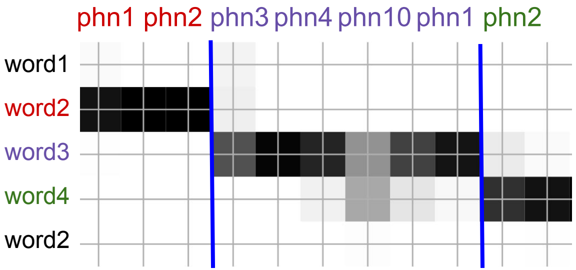
Experimental Settings

Task Pipeline

Encoder:
word1,word2,word3,word4...

Decoder:
phn1,phn2,phn3,phn4,phn10,phn1...

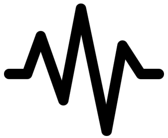
seq2seq system



Segmentation:
phn1phn2, phn3phn4phn10phn1, phn2

Alignment:
(phn1phn2, word2);
(phn3phn4phn10phn1, word3);
(phn2, word4)

Unit Discovery System

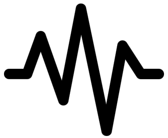


Task Pipeline: Data

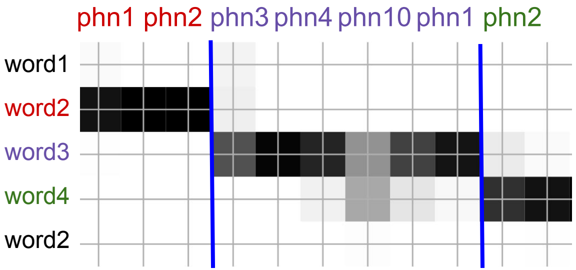
Encoder:
word1,word2,word3,word4...

Decoder:
phn1,phn2,phn3,phn4,phn10,phn1...

Unit Discovery System



seq2seq system



Segmentation:
phn1phn2, phn3phn4phn10phn1, phn2

Alignment:
(phn1phn2, word2);
(phn3phn4phn10phn1, word3);
(phn2, word4)

Corpora (3)

(MB-FR) Mboshi-French parallel corpus.
documentation dataset; tailored sentences
(EN-FR) English-French parallel corpus.
librispeech augmentation in French; noisy aligned information (filtered)

Data impact analysis

33K	EN-FR	
5K	EN-FR	MB-FR

Alignment quality analysis

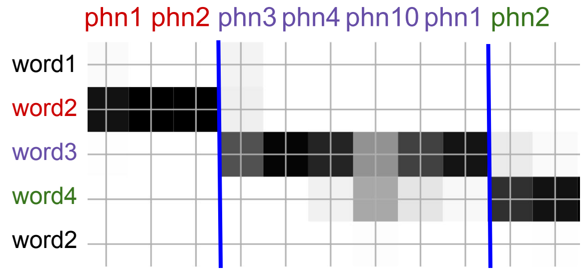
➔ In this work we use the true phones (as opposed to our Interspeech 2018 paper)

Task Pipeline: Models

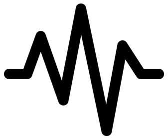
Encoder:
word1,word2,word3,word4...

Decoder:
phn1,phn2,phn3,phn4,phn10,phn1...

seq2seq system



Unit Discovery System



Segmentation:
phn1phn2, phn3phn4phn10phn1, phn2

Alignment:
(phn1phn2, word2);
(phn3phn4phn10phn1, word3);
(phn2, word4)

Sequence-to-Sequence (seq2seq) Models

RNN-based

Global attention.
(Bahdanau et al. 2015)

Attention mechanism creates context vectors.

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

$$\alpha_i^t = \text{softmax}(align(h_i, s_{t-1}))$$

Sequence-to-Sequence (seq2seq) Models

RNN-based

Global attention.
(Bahdanau et al. 2015)

Attention mechanism creates context vectors.

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

$$\alpha_i^t = \mathrm{softmax}(align(h_i, s_{t-1}))$$

Transformer

Multi-head attention using Scaled dot-product attention.
(Vaswani et al. 2017)

Attention is a mapping problem between key-value and query vectors.

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

$$h_i = Att(f_i(V), f_i(K), f_i(Q))$$

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

Sequence-to-Sequence (seq2seq) Models

RNN-based

Global attention.
(Bahdanau et al. 2015)

Attention mechanism creates context vectors.

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

$$\alpha_i^t = \text{softmax}(align(h_i, s_{t-1}))$$

Transformer

Multi-head attention using Scaled dot-product attention.
(Vaswani et al. 2017)

Attention is a mapping problem between key-value and query vectors.

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

$$h_i = Att(f_i(V), f_i(K), f_i(Q))$$

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

2D CNN-based

Pervasive attention.
(Elbayad et al. 2018)

Source elements are re-encoded considering the generated output.

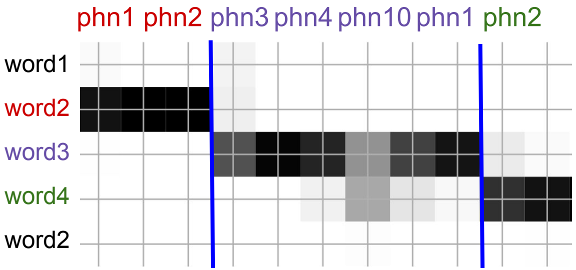
$$\begin{aligned} \alpha &= \text{softmax}\left(W_1 \tanh\left(H_L W_2\right)\right) \\ H_L^{\text{Att}} &= \alpha H_L . \end{aligned}$$

Task Pipeline: Evaluation (1)

Encoder:
word1,word2,word3,word4...

Decoder:
phn1,phn2,phn3,phn4,phn10,phn1...

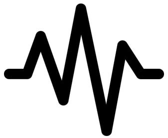
seq2seq
system



Segmentation:
phn1phn2, phn3phn4phn10phn1, phn2

Alignment:
(phn1phn2, word2);
(phn3phn4phn10phn1, word3);
(phn2, word4)

Unit Discovery
System



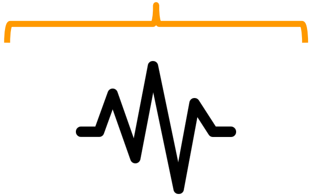
Task Pipeline: Evaluation (2)

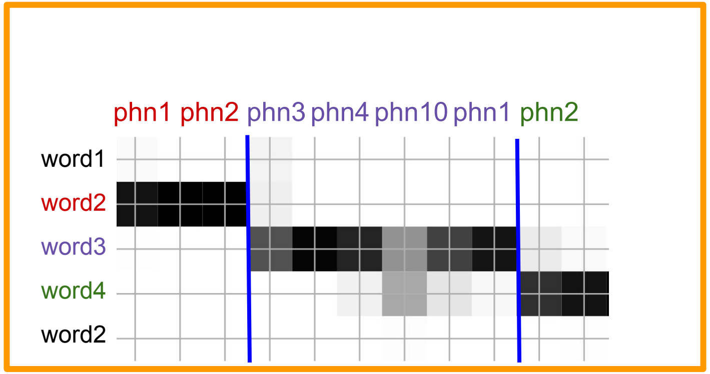
Encoder:
word1,word2,word3,word4...

Decoder:
phn1,phn2,phn3,phn4,phn10,phn1...

seq2seq system

Unit Discovery System



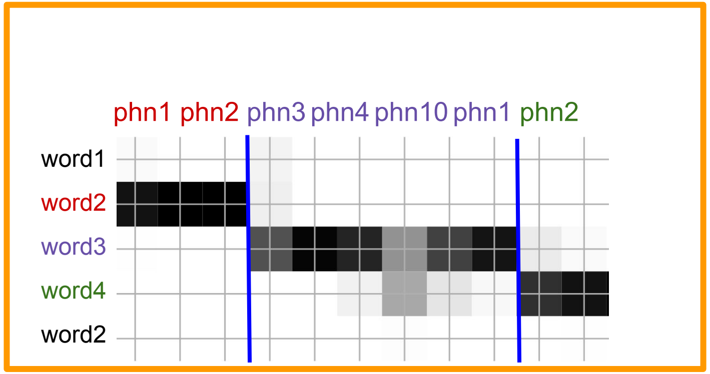


Segmentation:
phn1phn2, phn3phn4phn10phn1, phn2

Alignment:
(phn1phn2, word2);
(phn3phn4phn10phn1, word3);
(phn2, word4)

Intrinsic Evaluation

How do we evaluate alignment quality without having gold (word-level) alignment information?



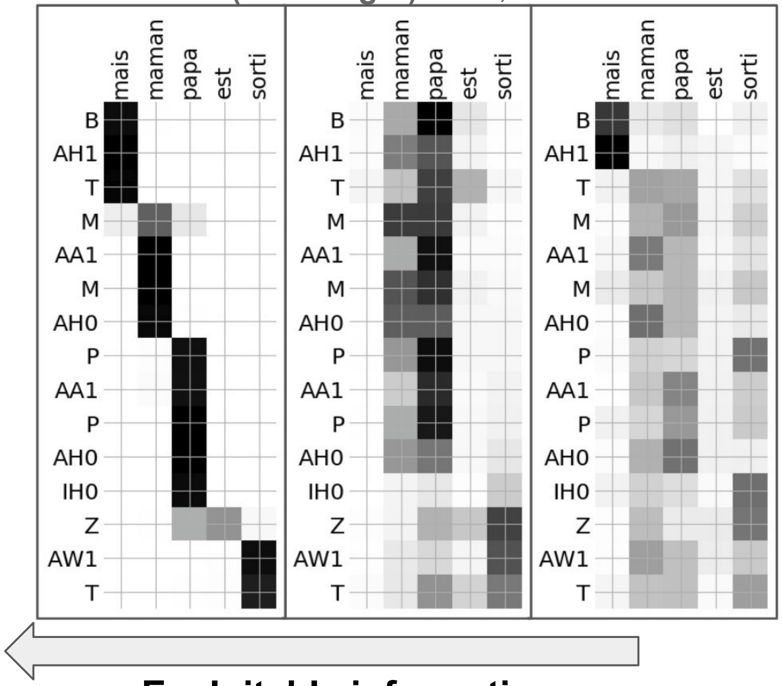
We use **Average Normalized Entropy** for assessing *alignment confidence.*

Average Normalized Entropy

Intuition: sharper alignments are more informative.

Soft-alignment probability matrix: one probability distribution per line (target symbol)

Matrix ANE (left to right): 0.11, 0.64 and 0.83.

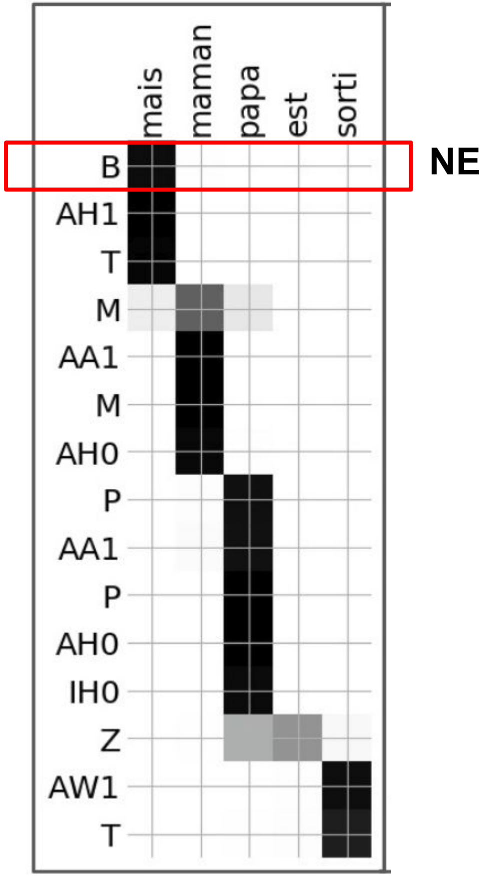


Exploitable information

Average Normalized Entropy

For every line in the matrix we compute normalized entropy (NE).

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

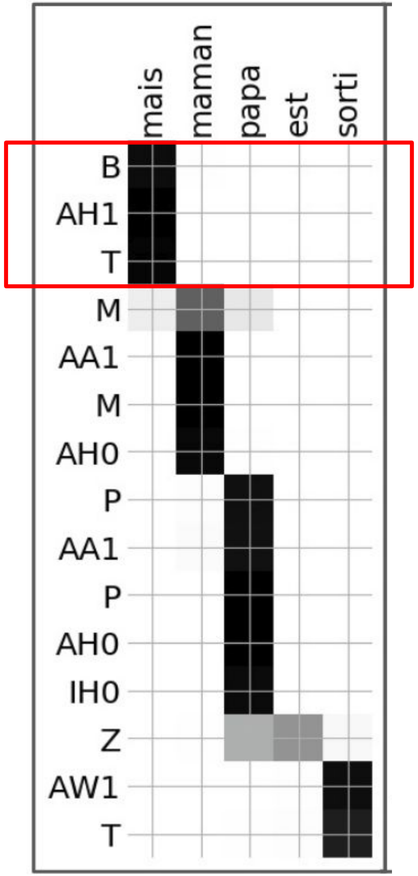


Average Normalized Entropy

For every line in the matrix we compute normalized entropy (NE). We average over sets of distributions.

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

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



Alignment

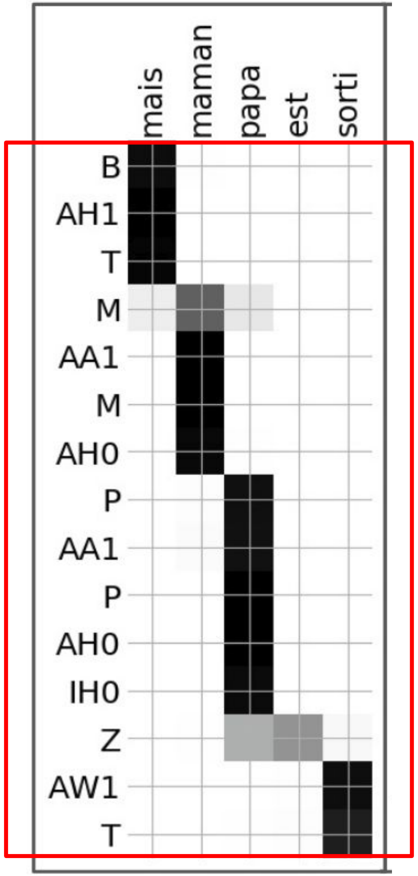
ANE

Average Normalized Entropy

For every line in the matrix we compute normalized entropy (NE). We average over sets of distributions.

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

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



Sentence

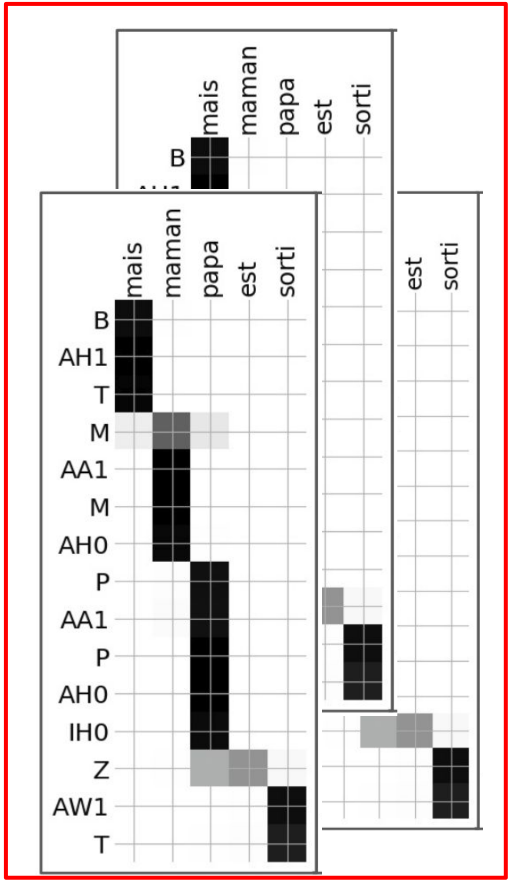
ANE

Average Normalized Entropy

For every line in the matrix we compute normalized entropy (NE).
We average over sets of distributions.

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

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

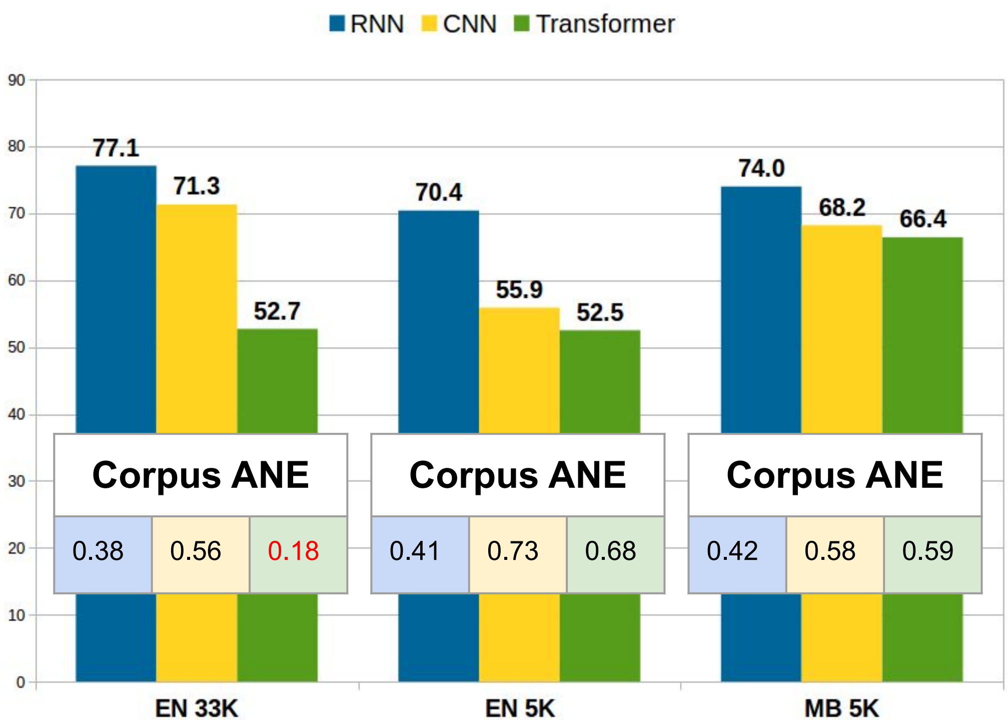


Corpus

ANE

Results

Unsupervised Word Segmentation Results

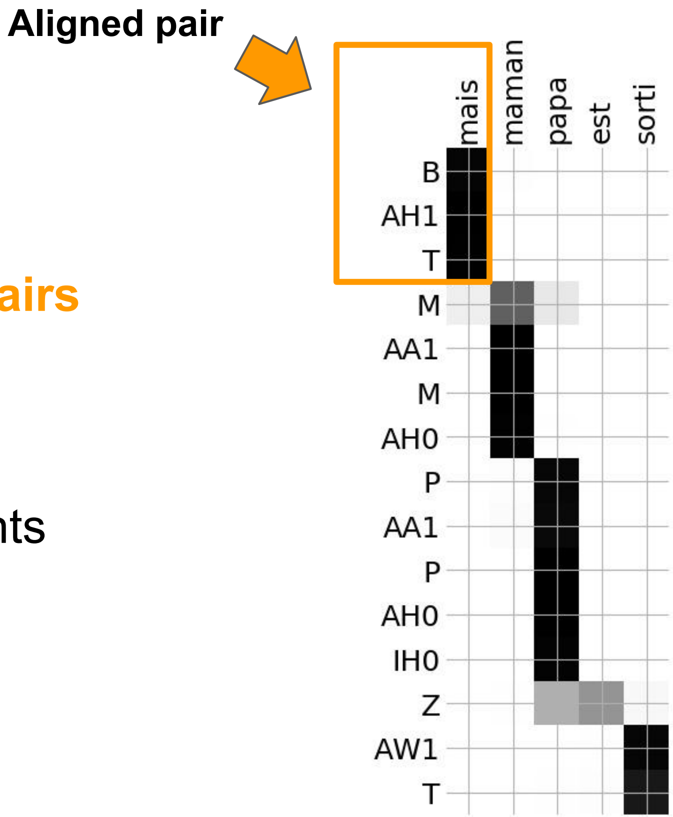


Boundary F-score Results. Results are the average of 5 runs.

- ➔ RNN-based model performed the best
- ➔ Models with lower **Corpus ANE** reached better segmentation results (strong negative Pearson's correlation verified for all models)

Experimental settings and corpora:
https://gitlab.com/mzboito/attention_study

ANE Application: Exploiting the Alignments



We accumulate ANE for all the
(discovered type, aligned information) pairs
discovered by our best 5K models
in the whole corpus

This allow us to rank discovered alignments
by their confidence.

Alignment ANE: Type Discovery Results

ANE over (discovered type, aligned information) pairs for the entire dataset

	EN 5K			MB 5K		
ANE	P	R	F	P	R	F
≤ 0.1	70.97	0.50	1.00	72.13	0.57	1.12

- **High-confidence alignments cover a small portion of the corpus, but have high precision**

Alignment ANE: Type Discovery Results

ANE over (discovered type, aligned information) pairs for the entire dataset

Confidence degree

ANE	EN 5K P	EN 5K R	EN 5K F	MB 5K P	MB 5K R	MB 5K F
$\leq$ 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**

- **Accepting a wider confidence window, we decrease precision results, but increase coverage**

Alignment ANE: Type Discovery Results

Alignment ANE can be used for filtering the resulting lexicon, increasing type discovery results

	EN 5K			MB 5K		
ANE	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

Alignment ANE: Type Discovery Results

- ➔ **Low ANE:** more frequently correct types, good alignment
- ➔ **High ANE:** more frequently incorrect types and alignments artifacts

	Phoneme Sequence	Grapheme	Aligned Information
1	SER1	sir	</S>
2	HHAH1SH	hush	chut
3	FIH1SHER0	fisher	fisher
4	KLER1K	clerk	clerc
5	KIH1S	kiss	embrasse
6	GRIH1LD	grilled	grilled
7	WUH1D	would	m’ennuierais
8	HHEH1LP	help	aidez
9	DOW1DOW0	dodo	dodo
10	KRAE1BZ	crabs	crabes

Top low alignment ANE pairs for EN5K.

	Phoneme Sequence	Grapheme	Aligned Information
1	AH0	a	convenablement
2	IH1	Not a word	ah
3	D	Not a word	riant
4	N	Not a word	obéit
5	YUW1	you	diable
6	IH1	Not a word	qu’en
7	AE1T	at	laquelle
8	Z	Not a word	bas
9	YUW1P	Not a word	</S>
10	L	Not a word	parfaitement

Top high alignment ANE pairs for EN5K.

Conclusion

Concluding...

For low-resource,

→ RNN-based model outputs the most easily exploitable soft-alignments

→ Transformer soft-alignment exploitation remains a challenge (what are the heads learning?[1,2])

[1]Michel, Levy. **Are Sixteen Heads Really Better than One?** NeurIPS 2019.
[2]Voita et al. ***Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned.*** ACL 2019.

Concluding...

For low-resource,

→ RNN-based model outputs the most easily exploitable soft-alignments

→ Transformer soft-alignment exploitation remains a challenge (what are the heads learning?[1,2])

We introduced Average Normalized Entropy (ANE) showing:

→ Its correlation to the segmentation scores

→ Its usefulness for filtering alignments

Thank you!

Questions?

Contact: marcely.zanon-boito@univ-grenoble-alpes.fr

UNIVERSITÉ Grenoble Alpes

Experimental settings and corpora:
https://gitlab.com/mzboito/attention_study

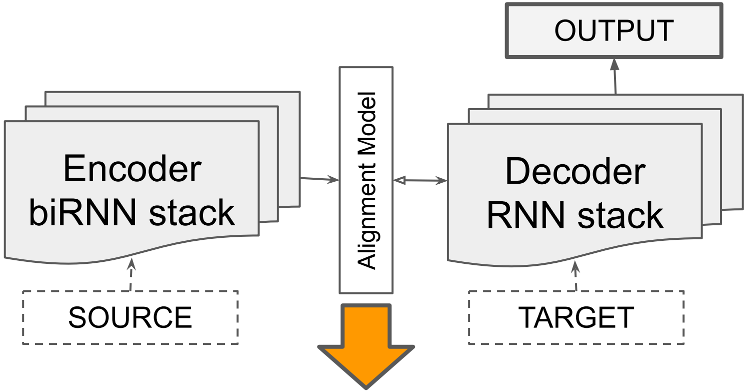
Empirical Evaluation of Sequence-to-Sequence Models for Word Discovery in Low-resource Settings

Marcely Zanon Boito[1], Aline Villavicencio[2,3] and Laurent Besacier[1]

[1]LIG, Univ. Grenoble Alpes, **France**
[2]CSEE, Univ. of Essex, **UK**
[3]INF, UFRGS, **Brazil**

Models: RNN

Global attention from Bahdanau et al. 2015



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

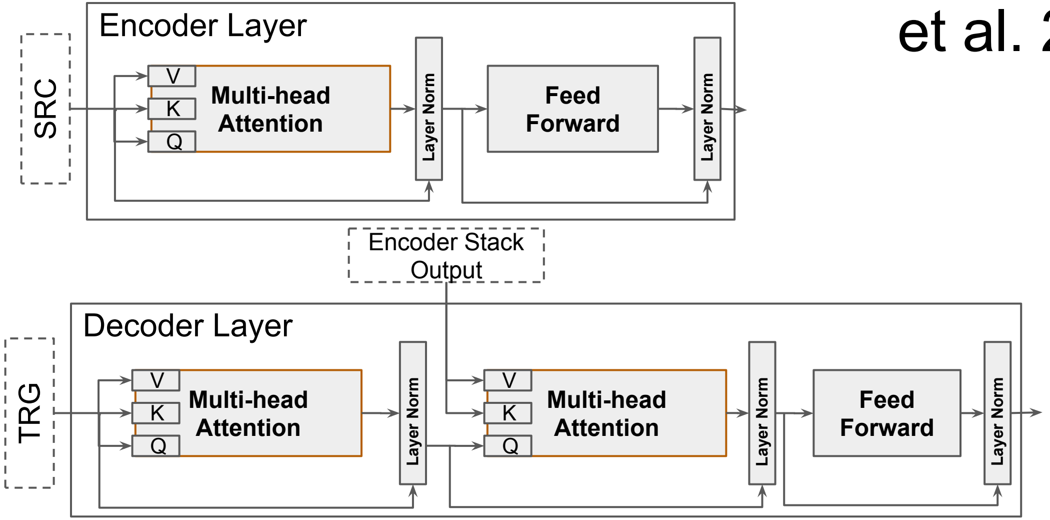
$$\alpha_i^t = \text{softmax}(align(h_i, s_{t-1}))$$

Attention appears at the form of a context vector for a decoder step *t*.

It is computed using the set of source annotations H and the last state of the decoder network s_{t-1} (translation context).

Models: Transformer

Multi-head attention from Vaswani et al. 2017



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

$$h_i = Att(f_i(V), f_i(K), f_i(Q))$$

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

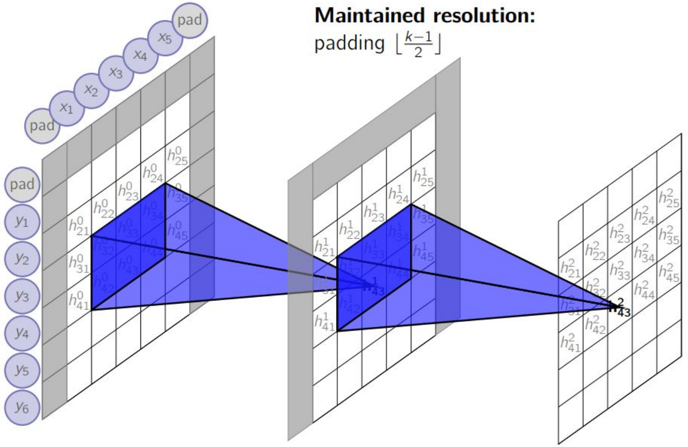
Attention is a mapping problem: given a pair of key-value vectors and a query vector, the goal is to compute the weighted sum of the key-values.

Weights are learned by compatibility functions between key-query pairs using Scaled dot-product (SDP) for each head in H.

Multi-head attention: SDP for several heads, which are then concatenated and projected again to yield the final result.

Models: CNN

Pervasive attention form Elbayad et al. 2018



$$\begin{aligned} \alpha &= \operatorname{softmax}\left(W_1 \tanh\left(H_L W_2\right)\right) \\ H_L^{\text{Att}} &= \alpha H_L . \end{aligned}$$

Source and target are encoded jointly, what acts an an attention-like mechanism since individual source elements are re-encoded as the output is generated.

Attention weight tensor α is computed from the last activation tensor H_L , to pool the elements of the same tensor along the source dimension.

Figure from https://elbayadm.github.io/