CLUZH at SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme Conversion

University of Zurich[UZH]

Peter Makarov and Simon Clematide
Institute of Computational Linguistics
makarov@cl.uzh.ch simon.clematide@cl.uzh.ch

Highlights

We adapt our SIGMORPHON 2018 system [2] to **Grapheme-to-Phoneme (G2P)**, a problem with largely disjoint input and output vocabularies.

박수 ↦ p a̲ k s̬ u

Before:

- Neural transducer with copy edit
- Trained with imitation learning [1, IL] and ED(target, prediction) + ED(input, prediction) as loss.

Now:

- Neural transducer with **substitution edits**
- Trained with IL and ED(target, prediction) + **Stochastic ED$_\phi$(input, prediction)** as loss. Stochastic ED is a WFST with parameters ϕ, which we learn from data.

Model

Just like a WFST, the model monotonically transduces input string **x** into output string **y** by a sequence of edits **a**.

- Edits: INS[y] and **SUB[y] for $y \in \Sigma_y$**, DEL, COPY
- Features: full input context, full edit history

$$\mathbf{s}_t = \text{LSTM}\big(\mathbf{c}_{t-1}, [E(a_{t-1}); \mathbf{h}_i]\big) \quad (1)$$

  + $\mathbf{c}_{t-1}$ is the previous cell state and output,
  + $E(a_{t-1})$ is the embedding of previous edit, and
  + $\mathbf{h}_i$ is the biLSTM encoding of input character x_i.

- Probabilities of edits:

$$P(a_t \mid \mathbf{a}_{<t}, \mathbf{x}, \theta) = \text{softmax}(\mathbf{w} \cdot \mathbf{s}_t) \quad (2)$$

Example: Model Execution

G	a	ε	b	ε	s	ε	e	n	t
	|	|	|	|	|	|	|	|	|
Edits	COPY	INS[•]	SUB[p]	INS[•]	COPY	INS[•]	SUB[ɑ]	SUB[˜]	DEL
	|	|	—	|	|	|	—	|	|
P	a	•	p	•	s	•	ɑ	˜	ε

• = whitespace

Stochastic Edit Distance

Our 2018 IL objective ranks edit action sequences by action cost: ED(input, prediction). This addresses spurious ambiguity as multiple edit action sequences achieve the same ED(target, prediction).

For G2P, we use Stochastic Edit Distance [3, SED], a probabilistic generalization of ED. We learn edit weights ϕ from data with EM.

IL training: Maximize likelihood of edits that are optimal under the loss in any configuration (input, attention pointer, prediction so far, and target).

We find optimal edits on the training set using the **SED expert policy**:

1. Find **permissible edits**: They do not increase ED(target, future prediction)

   Each permissible edit corresponds to a **target suffix** (=what needs to be completed of the target).

2. **Score** permissible edits **using SED**:
   (a) Execute edit
   (b) Edit cost-to-go = edit cost under SED
   (c) Edit cost-to-go += cost of Viterbi path in SED for (input suffix, target suffix).
3. Optimal edits attain lowest cost-to-go

Example: SED policy

- input **x** = abject
- target **y** = a•b•ʒ•ɛ•k•t
- attention x_4 = e, and
- imperfect prediction so far $\hat{\mathbf{y}}_{1:7}$ = a•b•ʒ•e

1. Permissible edits: SUB[ɛ], INS[ɛ], DEL, SUB[•], INS[•]

   Target suffixes: e.g. "ɛ•k•t" for SUB[ɛ], or "•k•t" for SUB[•]

2. Score permissible edits:
   (a) Execute edit: e.g. SUB[ɛ] writes ɛ and moves the attention to x_5 = c
   (b+c) Compute cost-to-go: **15.3 for SUB[•]**, 17.7 for SUB[ɛ], 21.1 for INS[ɛ], 17.3 for DEL, and 17.3 for INS[•]

Results: Overall 2nd Best System

	CLUZH		ENS.		CLUZH WER AVG				LSTM	TF	BEST BY OTHERS			
LNG	WER	PER	#C	#D	WER	±	Δ,%	⊥	WER	WER	WER	Δ,%	PER	Δ,%
ady	27.11	6.27	0	11	30.32	1.97	-12	16.89	28.00	28.44	**24.67**	9	**5.76**	8
arm	**12.22**	**2.82**	0	11	14.73	0.76	-21	8.89	14.67	14.22	12.67	-4	2.91	-3
bul	23.33	**4.70**	0	11	30.81	2.78	-32	13.78	31.11	34.00	**22.22**	5	**4.70**	0
dut	14.44	2.51	9	2	18.30	1.44	-27	9.33	16.44	15.78	**13.56**	6	**2.36**	6
fre	6.89	1.56	2	9	8.12	0.54	-18	3.56	6.22	6.89	**5.11**	26	**1.16**	26
geo	27.33	4.83	0	11	29.11	0.86	-7	8.89	26.44	28.00	**24.89**	9	**4.57**	5
gre	16.44	2.68	11	0	19.60	1.80	-19	7.33	18.89	18.89	**14.44**	12	**2.42**	10
hin	**5.11**	**1.20**	0	11	7.13	0.55	-40	2.67	6.67	9.56	**5.11**	0	**1.20**	0
hun	**4.00**	1.02	0	11	4.77	0.60	-19	2.89	5.33	5.33	**4.00**	0	**0.92**	10
ice	**9.11**	1.90	0	11	10.00	0.53	-10	5.78	10.00	10.22	**9.11**	0	**1.83**	4
jpn	6.00	1.58	0	11	7.19	0.30	-20	4.89	7.56	7.33	**4.89**	19	**1.16**	27
kor	28.44	4.88	0	11	28.26	1.39	1	11.78	46.89	43.78	**24.00**	16	**4.05**	17
lit	**18.67**	**3.27**	0	11	21.54	0.82	-15	14.22	19.11	20.67	**18.67**	0	3.38	-3
rum	11.33	2.68	0	11	13.66	1.11	-21	7.11	10.67	12.00	**9.78**	14	**2.23**	17
vie	1.56	0.35	0	11	1.60	0.21	-2	0.89	4.67	7.56	**0.89**	43	**0.27**	23
AVG	14.13	2.82	1.5	9.5	16.34	1.05	-16	7.93	16.84	17.51	**12.93**	8	**2.59**	8

Table 1: Overview of the test results. Δ gives relative error difference compared to our submission CLUZH. #C=number of NFC models in the ensemble. #D=number of NFKD models in the ensemble. CLUZH WER AVG=average WER, standard deviation, and relative error difference of the average computed over individual models. ⊥=lower-bound on WER: correct if predicted by any individual model. LSTM=official seq2seq LSTM baseline. TF=official seq2seq Transformer baseline. BEST BY OTHERS=best results of other systems for each language.

Qualitative Error Analysis

ady	’/ɛ/17	ə•/ɛ/9	ʃ/ʂ/8	ɛ/•ə/7	j•/ɛ/6
arm	ɔ/o/17	ɛ/ə•/12	◌͡/•/12	ə•/ɛ/3	t/d/3
bul	r/ɾ/26	o/ɔ/22	ə/a/14	a/ə/12	◌̈/ɛ/9
dut	ə/ɛ/9	ɛ/j•/4	aː/ɑ/4	eː/ə/4	ə/eː/3
fre	ɛ/•ɑ̃/2	ɛ/•s/2	a/ɑ/2	o/ɔ/2	w/ɔ/2
geo	ɪ/i/103	i/ɪ/48	χ/x/5	ɣ/ʁ/4	ʁ/ɣ/3
gre	ɾ/r/27	o/ɔ/19	r/ɾ/15	e/ɛ/9	j̩/i/3
hin	ɛ/ə•/10	ə•/ɛ/5	ɛ/•ə/2	ɛː/ə/2	ɛ/◌̯•/2
hun	ʃ/ʒ/3	ɛ/ː/3	eː/i•n•t/1	ɛ/ɱ•v•/1	m/eʲ/1
ice	ː/ɛ/11	ɛ/ː/9	t•/ɛ/4	v/f/3	ɛ/◌̥/3
jpn	ɛ/◌̊/8	ɛ/◌̥/6	ɛ/ː/3	ː/•ɯᵝ/2	ː/•o̥/2
kor	ɛ/ː/72	ː/ɛ/18	ɘː/ʌ̹/11	ʑ/ɕ/4	ʌ/ɘː/4
lit	ɛ/◌̤/15	n/ŋ/14	ʁ/aː/12	ː/ɛ/8	ʲ/ɛ/7
rum	◌͡/•/8	•/◌͡/8	e̯/j/6	r/ɾ/5	ʲ/•i/4
vie	ɛ/ː/2	⊣•/ɛ/1	ɛ/e•/1	w•/ɛ/1	˥•/ɛ/1

Table 2: Five most frequent errors per language. Notation: prediction / gold / error frequency. Computed with the help of ISRI Analytic Tools for OCR Evaluation.

Setup

Parameters: input character & action embeddings of size 100 and one-layer LSTMs with hidden-state size 200.

Training: maximum of 60 epochs with a patience of 12, mini-batches of size 5. Training takes 4 minutes per epoch on CPU (DyNet).

References

[1] Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume III, and John Langford. Learning to search better than your teacher. In *ICML*, 2015.
[2] Peter Makarov and Simon Clematide. UZH at CoNLL-SIGMORPHON 2018 shared task on universal morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, 2018.
[3] Eric Sven Ristad and Peter N Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

Acknowledgements: This work was supported by Swiss National Science Foundation Grant No. CRSII5_173719.