The CMU-LTI submission to the SIGMORPHON 2020 Shared Task 0: Language-Specific Cross-Lingual Transfer

NEULAB

Nikitha Murikinati and Antonios Anastasopoulos

{nmurikin,aanastas}@andrew.cmu.edu

Carnegie Mellon University Language Technologies Institute

Highlights

Morphological Inflection is the task where, given a lemma, e.g.

aguar

and a set of morphological tags, e.g.

`V; PRS; 2; PL; IND;`

one has to generate the correctly inflected form, e.g.

aguà

In low-resource settings this task is still very challenging.

We combine several techniques:

1. a novel two-step attention for the decoder
2. data hallucination
3. multi-tasking with a simple copying task
4. cross-lingual transfer from **multiple related** languages

and achieved state-of-the-art results over 44 test languages (from the SIGMORPHON 2019 challenge), with a gain of more than 15 points over the baseline.

In the SIGMORPHON 2020 Task 0 shared task, our additions were:

1. Add transliterated/romanized transfer language data for related language pairs that nevertheless use different scripts:
   - Classical Syriac (Arabic, Hebrew)
   - Maltese (Italian, Hebrew)
   - Oromo (Arabic, Hebrew)
   - Bengali (Sanskrit, Hindi)
   - Tajik (Farsi)
   - Pashto (Farsi)
2. create language specific transfer models using related languages **only** for low-resource settings, e.g.:
   - Ladin (Friulian)
   - Ludian (Karelian, Veps)

Results:

Ranked 20th among 31 systems, with non-optimized LSTM-based systems.

Take-away:

The top-3 systems of the shared task offer much better solutions, which however should be able to be improved upon using language-specific approaches.

Two-Step Attention for Disentangled Inputs

First, encode the tag sequence and the lemma:

$$\mathbf{h}^x_n = \mathrm{enc}^{\mathrm{x}}(\mathbf{h}^x_{n-1}, x_n) \quad \text{and} \quad \mathbf{h}^t_m = \mathrm{enc}^{\mathrm{t}}(\mathbf{T}).$$

For each decoding step,

a) get context from tag attention

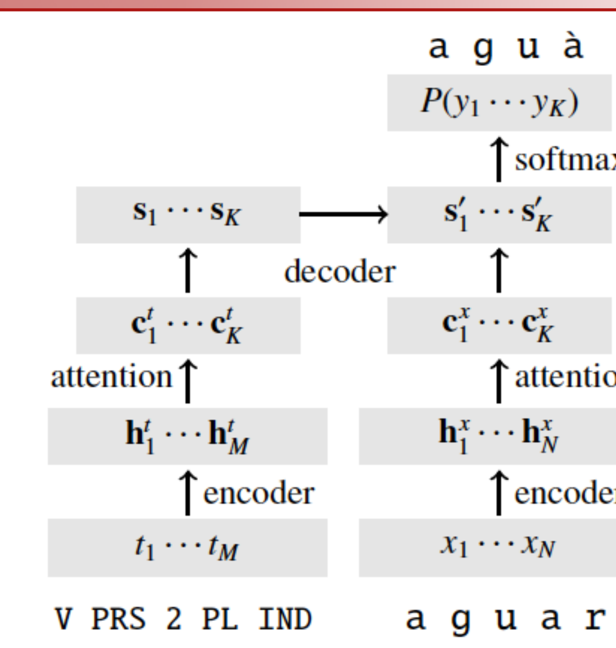
b) obtain a tag-informed decoder state

c) attend over lemma

d) produce output character

$$\mathbf{s}_k = \mathbf{s}'_{k-1} + \mathbf{c}^t_k$$
$$\mathbf{s}'_k = \mathrm{dec}(\mathbf{s}'_{k-1}, \mathbf{c}^x_k, y_{k-1})$$
$$P(y_k) = \mathrm{softmax}(\mathbf{s}'_k).$$

$$\mathbf{c}^x_k = \left[\textstyle\sum_n \alpha^x_{kn}\mathbf{h}^x_n\right] \qquad \mathbf{c}^t_k = \left[\textstyle\sum_m \alpha^t_{km}\mathbf{h}^t_m\right]$$



Additional Biases

1. encourage monotonic attention: use an additional copying task (see training regime below)

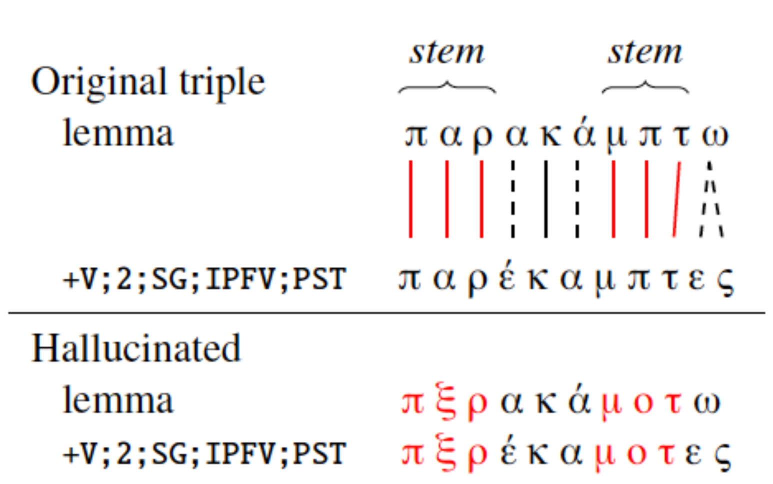
2. encourage attention coverage of the two sources: $-\lambda \parallel \Sigma_j a^t_{jm} - \mathbb{I} \parallel_2$ $\quad -\lambda \parallel \Sigma_j a^x_{jn} - \mathbb{I} \parallel_2$

3. Language discriminator over the encoder outputs (with gradient reversal): $y_l = \mathrm{softmax}(\mathrm{MLP}(\mathbf{h}^x_N))$

Data Hallucination

1. Find a "stem"-like region based on character alignment that remains unchanged
2. Randomly replace the inside characters

(example from Greek)



Cross-Lingual Training Regime

1. Train only on copying task over all languages
   - large batch size and learning rate
2. Train on both inflection (80%) and copying (20%) tasks for all languages
   - upsample the low-resource language
   - learning rate decay and restart the optimizer
3. Train only on the test language inflection task
   - small batch size
   - scheduled sampling

Results

Language	Accuracy	Language	Accuracy	Language	Accuracy	Language	Accuracy
aka	**99.1**	fas	96.2	lld	97.7	sna	**100.0**
ang	75.4	fin	97.3	lud	**53.7**	sot	**100.0**
ast	91.4	frm	98.8	lug	90.6	swa	**100.0**
aze	78.5	frr	85.5	mao	**69.0**	swe	95.4
azg	89.0	fur	98.3	mdf	92.7	syc	91.6
bak	97.4	gaa	**100.0**	mhr	90.8	tel	**94.9**
ben	98.6	glg	97.4	mlg	**100.0**	tgk	**93.8**
bod	**84.7**	gmh	90.1	mlt	88.7	tgl	64.0
cat	97.5	gml	**60.8**	mwf	70.3	tuk	85.4
ceb	**84.7**	gsw	84.9	myv	93.0	udm	97.5
cly	81.0	hil	92.4	nld	97.5	uig	91.9
cpa	83.5	hin	98.4	nno	74.2	urd	36.3
cre	44.9	isl	95.3	nob	**75.1**	uzb	51.5
crh	97.2	izh	80.8	nya	**100.0**	vec	98.8
ctp	50.2	kan	75.1	olo	91.5	vep	79.3
czn	**81.3**	kaz	88.5	ood	**79.0**	vot	77.2
dak	89.7	kir	88.4	orm	93.6	vro	**57.3**
dan	72.3	kjh	**98.8**	ote	97.0	xno	**90.2**
deu	92.8	kon	**98.1**	otm	97.4	xty	90.2
dje	**100.0**	kpv	95.9	pei	71.2	zpv	**82.9**
eng	96.5	krl	95.0	pus	68.6	zul	**89.7**
est	93.5	lin	**100.0**	san	92.6		
evn	55.0	liv	93.1	sme	97.9		

Table 1: Accuracy of our system on every language. We **highlight** the languages where our system was statistically equal to the best system (with $p < 0.005$).