Comparing Writing Systems with Multilingual Grapheme-to-Phoneme and Phoneme-to-Grapheme Conversion

University of the Basque Country
Language Analysis and Processing
Computational Morphology
Julen Etxaniz

Introduction

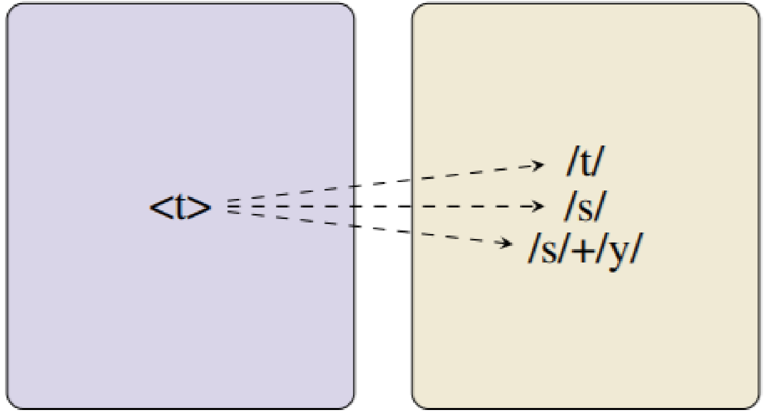
- Compare orthographic depth of languages
- Shallow (transparent) orthographies (Spanish, Basque)
  - One-to-one correspondence between graphemes and phonemes
- Deep (opaque) orthographies (English, French)
  - Less direct correspondence and more irregular words
- Asymmetric reading and writing difficulty (French)
- Learn character-level seq2seq models (transformer)
- G2P direction suggests difficulty of reading
- P2G direction suggests difficulty of writing

Introduction

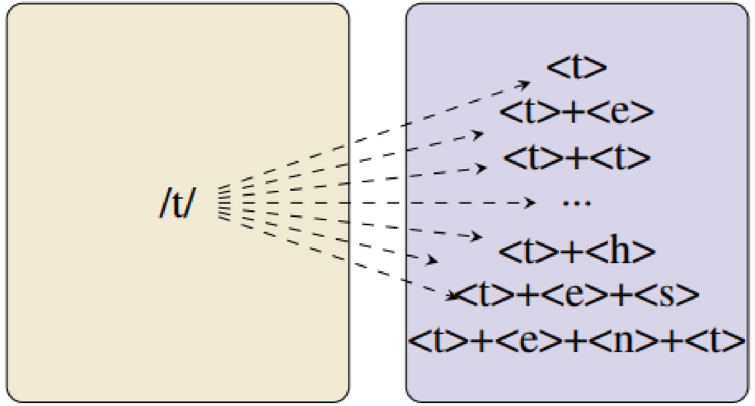
- French examples

Graphemes	Phonemes	Graphemes	Phoneme
ablatif	ablatif	t	t
abolition	abɔlisjɔ̃	t	s
acquitter	akite	tt	t
auguste	ogyst	te	t
athlète	atlɛt	th	t
actes	akt	tes	t
assistent	asist	tent	t

G2P



P2G



Data

- The SIGMORPHON 2021 G2P medium-resource data
- 10,000 examples of 10 languages
- Split into training (80%), development (10%) and testing (10%)
- Extracted from the English Wiktionary using WikiPron
- Allows to compare results with baseline
- Quality assurance to fix inconsistencies
- Fair comparison between languages

Preprocessing

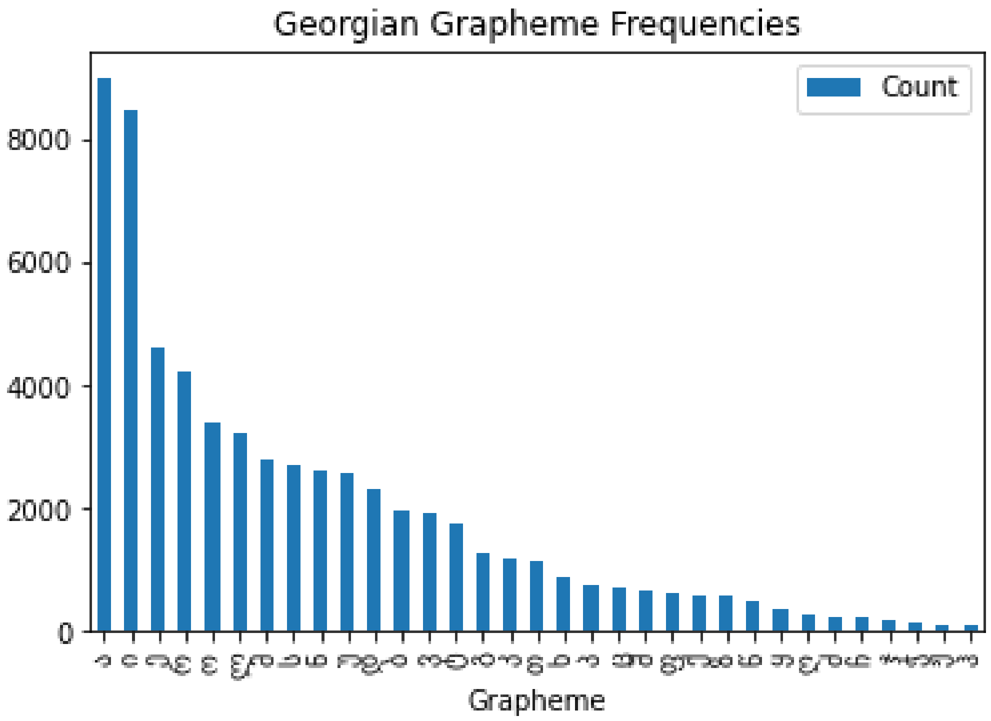
- Tokenize graphemes and phonemes
- Minimum of 5 occurrences of each token
- Remaining tokens mapped to unknown
- Calculate grapheme and phoneme frequencies
- Compare unique and average counts
- Extract some clues about results
- Georgian has the same amount
- French and Korean have different counts

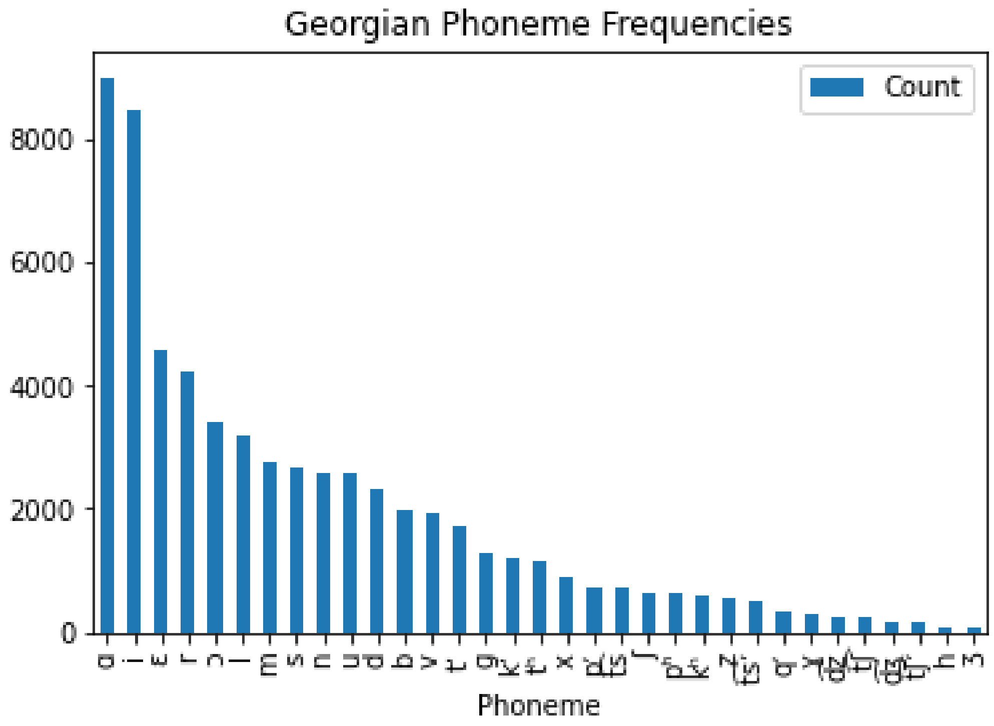
Preprocessing

Language	Code	Unique G	Unique P	Average G	Average P
Armenian (Eastern)	arm_e	38	41	7.09	7
Bulgarian	bul	29	45	8.44	8.64
Dutch	dut	30	45	7.71	6.99
French	fre	37	37	7.52	5.75
Georgian	geo	33	33	7.74	7.74
Serbo-Croatian (Latin)	hbs_latn	27	61	7.47	7.36
Hungarian	hun	34	61	7.65	7.18
Japanese (Hiragana)	jpn_hira	76	64	4.21	6.56
Korean	kor	559	60	2.58	6.54
Vietnamese (Hanoi)	vie_hanoi	89	49	5.81	7.51
Average	average	95.2	49.6	6.62	7.13

Preprocessing

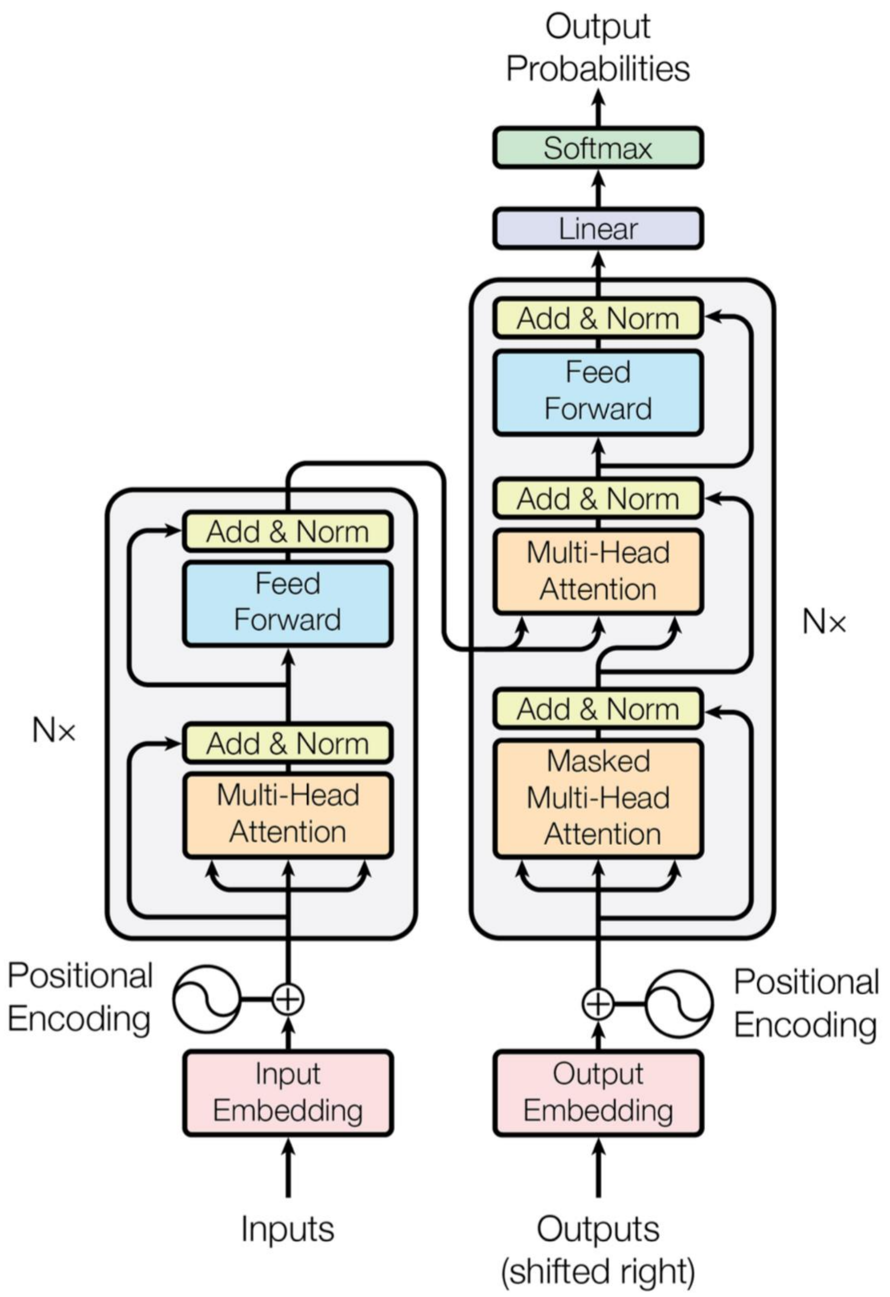
- Georgian: same grapheme and phoneme frequencies





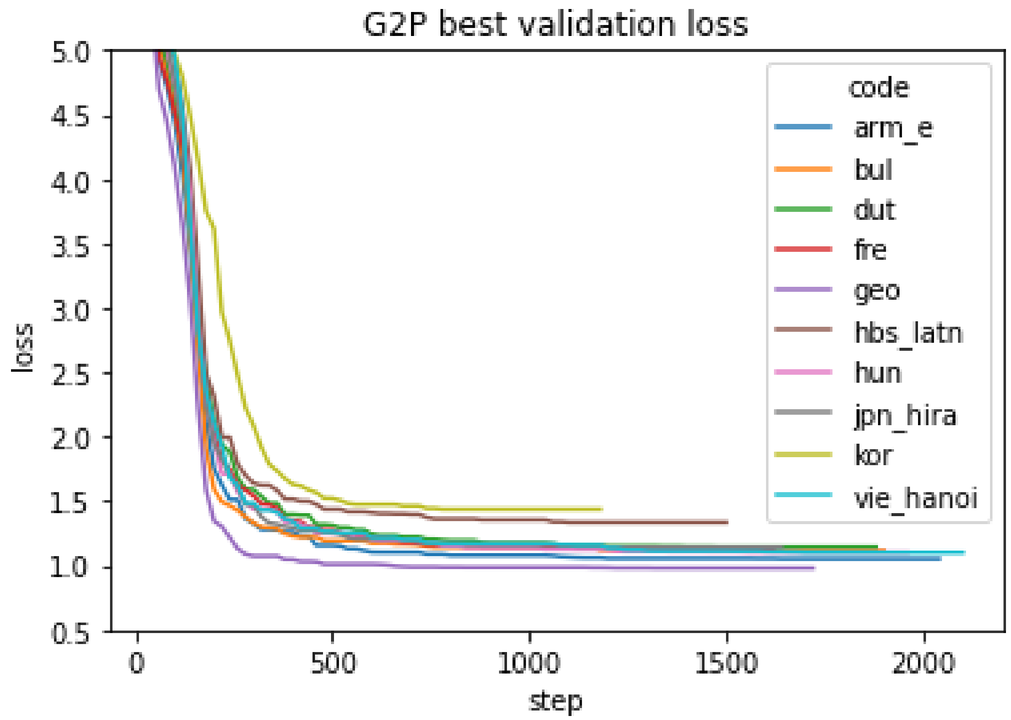
Training

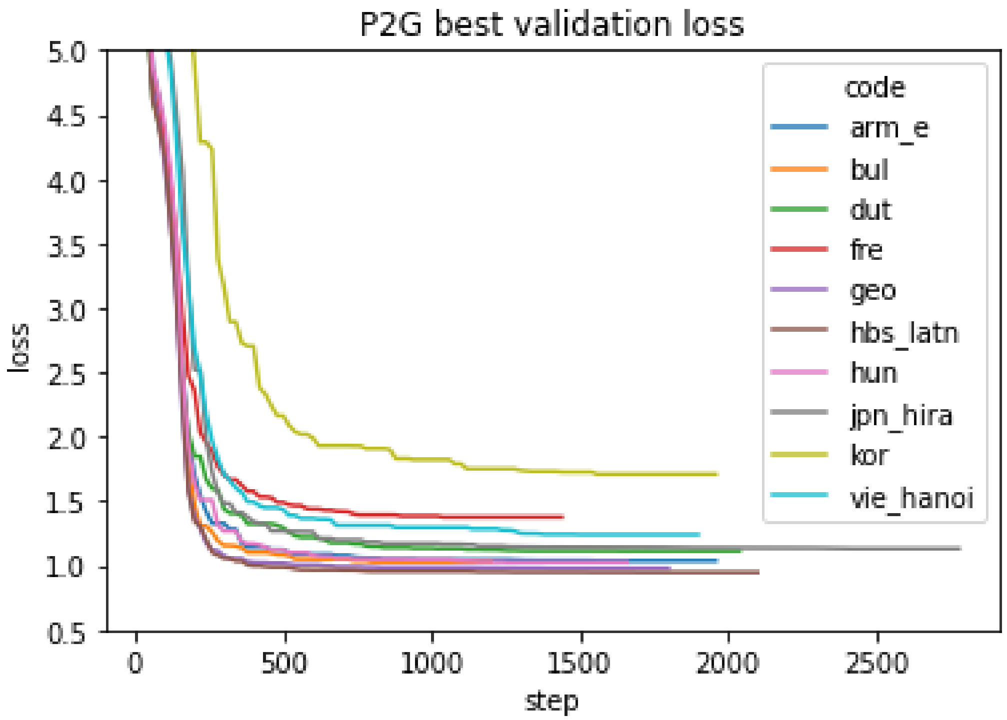
- Fairseq to train small transformer model
- 4 encoder-decoder layers
- 4 self-attention heads
- Embedding size is 256
- Hidden size of feed-forward layer 1024
- Around 7.4M parameters
- Batch size 400
- Learning rate 0.001
- Dropout rate 0.1
- Early stopping 20 epochs



Training

- Monitor using TensorBoard
- Biggest improvement in the first 500 steps





Testing

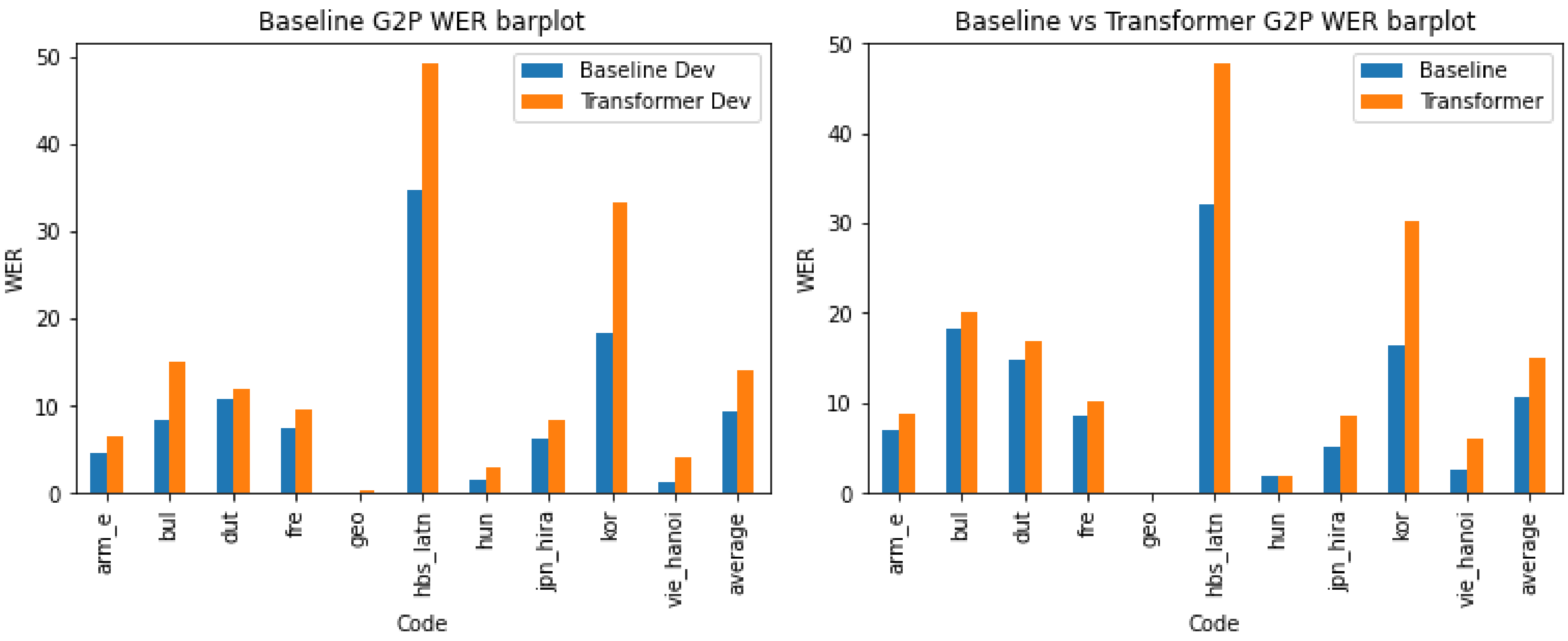
- Best checkpoint of each model
- Beam search with a size of 5
- Train, Dev and Test scores
- Word error rate (WER) metric
- Percentage of wrong word predictions

Results – G2P Baseline

Code	Baseline Dev	Transformer Dev	Baseline Test	Transformer Test
arm_e	4.50	6.5	7.00	8.90
bul	8.30	14.9	18.30	20.10
dut	10.80	12.0	14.70	16.90
fre	7.40	9.5	8.50	10.20
geo	0.00	0.3	0.00	0.10
hbs_latn	34.70	49.2	32.10	47.70
hun	1.50	2.8	1.80	1.90
jpn_hira	6.20	8.4	5.20	8.50
kor	18.40	33.4	16.30	30.20
vie_hanoi	1.30	4.0	2.50	6.00
average	9.31	14.1	10.64	15.05

Results – G2P Baseline

- Worse results, but good enough, ranks maintained

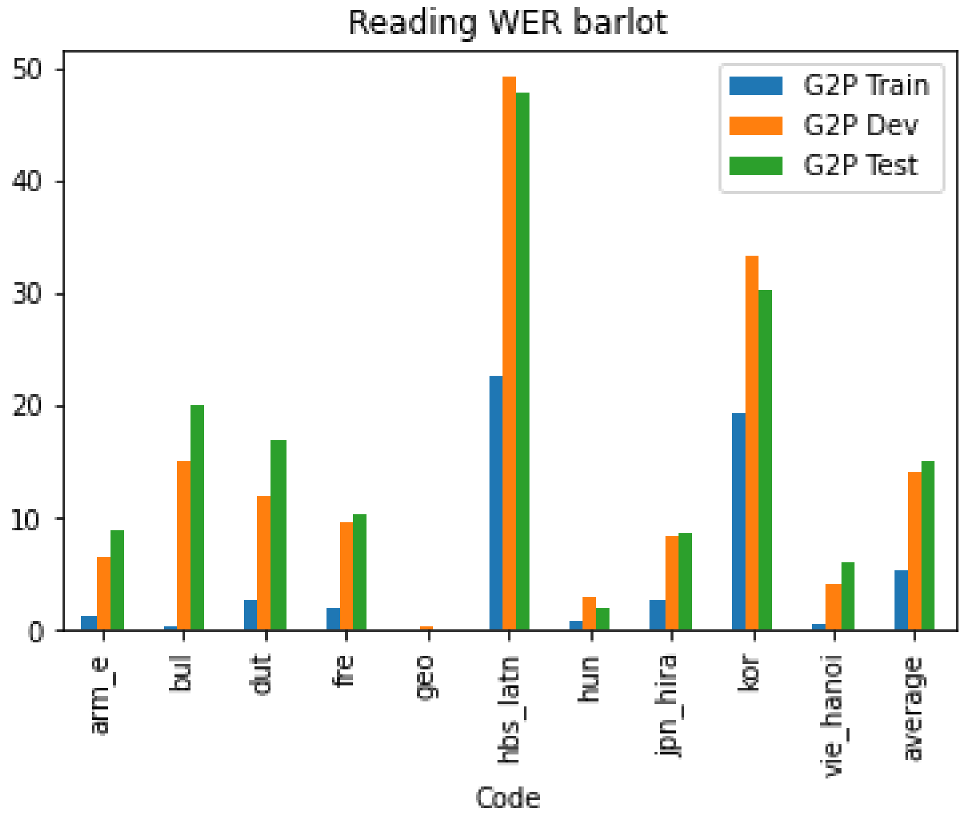


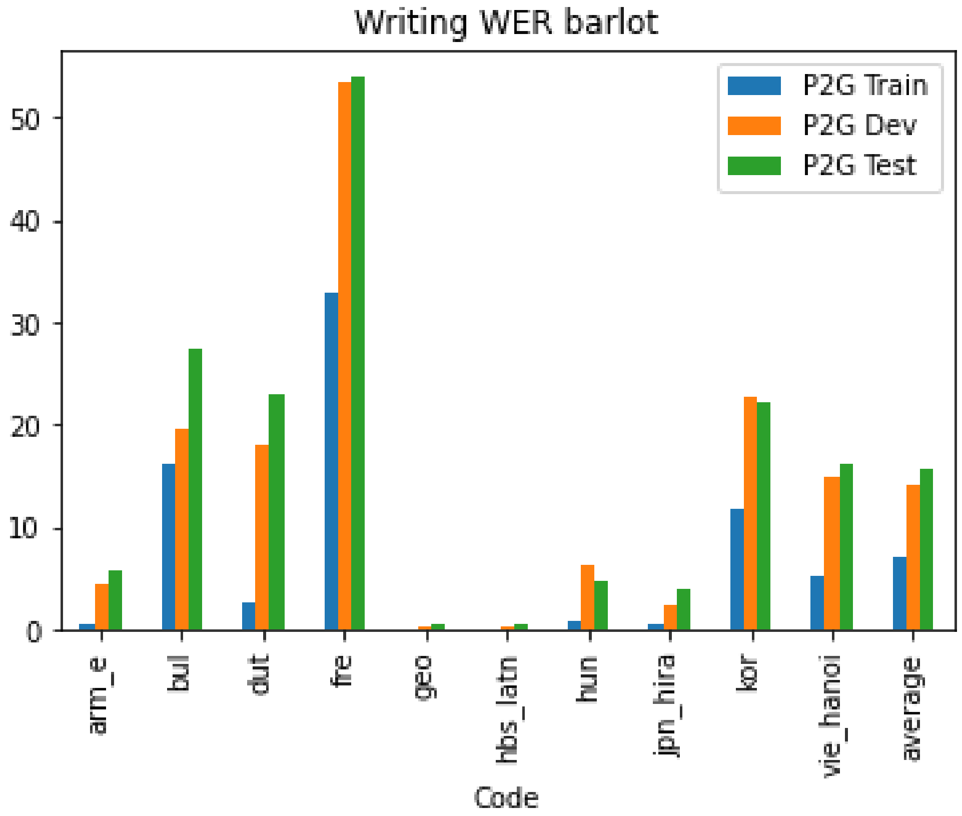
Results – G2P and P2G

Code	G2P Train	G2P Dev	G2P Test	P2G Train	P2G Dev	P2G Test
arm_e	1.16	6.5	8.90	0.57	4.40	5.70
bul	0.34	14.9	20.10	16.25	19.70	27.30
dut	2.59	12.0	16.90	2.67	18.00	23.00
fre	2.04	9.5	10.20	32.96	53.50	54.00
geo	0.06	0.3	0.10	0.12	0.30	0.50
hbs_latn	22.56	49.2	47.70	0.09	0.20	0.60
hun	0.64	2.8	1.90	0.84	6.20	4.80
jpn_hira	2.74	8.4	8.50	0.59	2.40	3.90
kor	19.27	33.4	30.20	11.84	22.80	22.10
vie_hanoi	0.44	4.0	6.00	5.35	15.00	16.30
average	5.18	14.1	15.05	7.13	14.25	15.82

Results – G2P and P2G

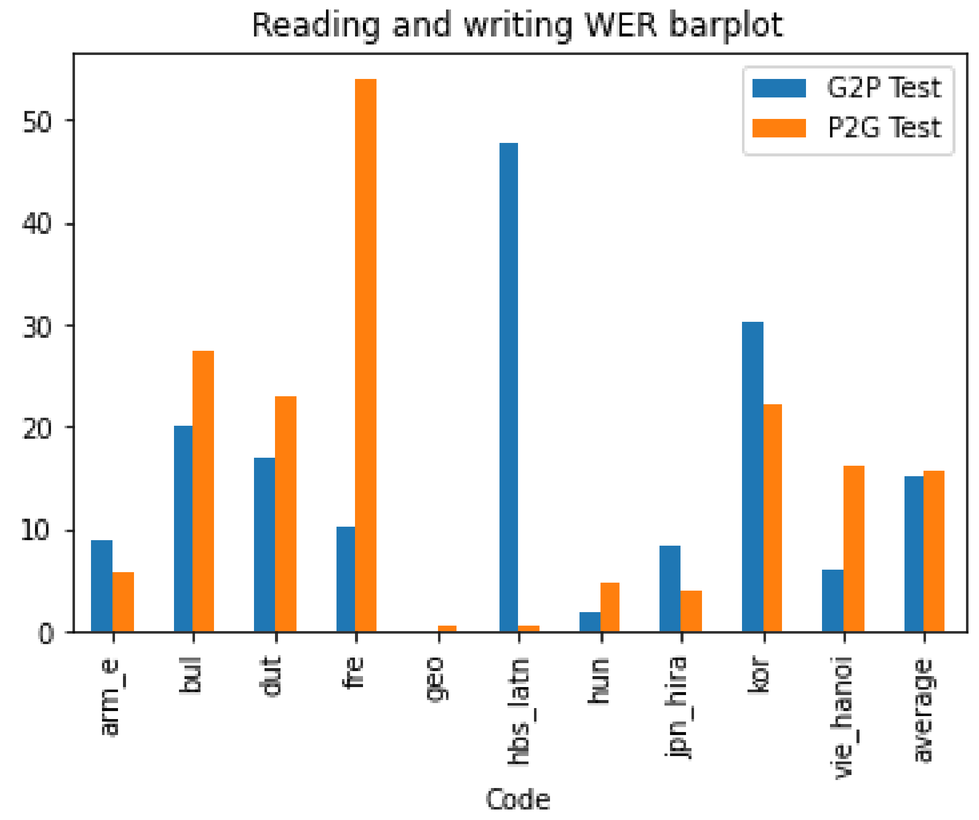
- Train vs Dev vs Test | G2P vs P2G | Language vs Language

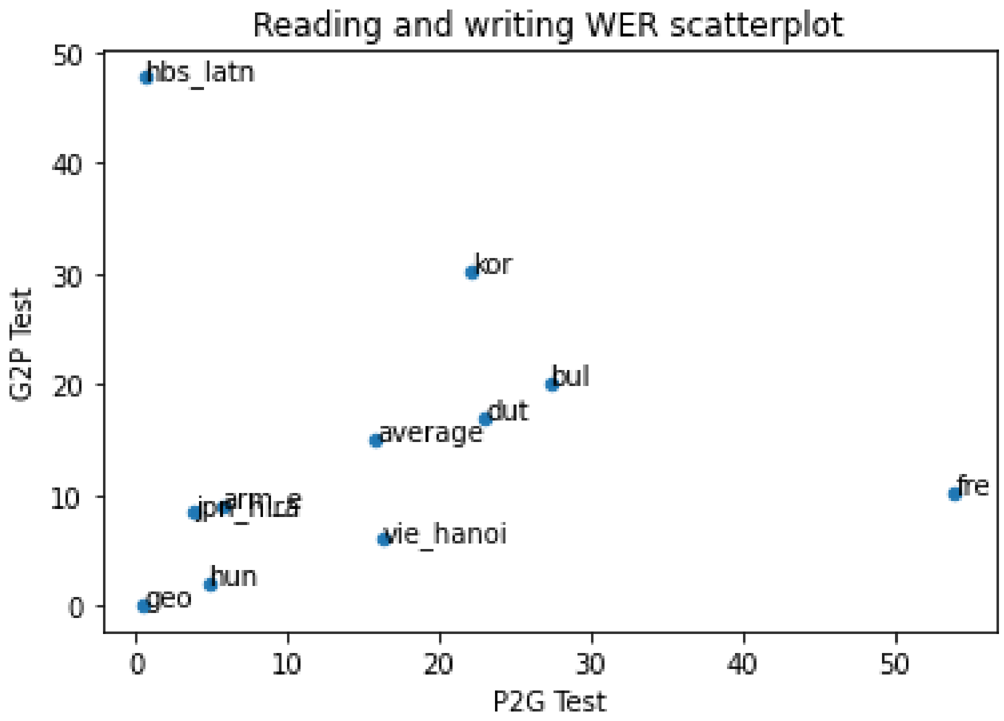




Results – G2P and P2G

- Rank and group languages based on reading and writing scores





Conclusions

- Results similar to related works
- A recent work used a single model for 17 languages
- Differences mainly due to inconsistencies in data
- These methods are good to estimate orthographic tranparency
- Fix inconsistencies to get more accurate results
- Apply method to more languages
- Limitations for smaller languages

Thank you!