Transfer Learning of Slavic Syllabification for Hyphenation Patterns

Ondřej Sojka

Faculty of Informatics, Masaryk University

October 16, 2024

Contents

Section 1

Why this problem

"Typographical prowess lies not in the ostentatious deployment of extravagant lexemes, but rather in the discerning mastery of the elegant harmony that interweaves characters, glyphs, and spaces, where the judicious orchestration of hyphenation serves as an exquisite testament to the printer's art." – not Edward Tufte

"Typographical prowess lies not in the ostentatious deployment of extravagant lexemes, but rather in the discerning mastery of the elegant harmony that interweaves characters, glyphs, and spaces, where the judicious orchestration of hyphenation serves as an exquisite testament to the printer's art." – not Edward Tufte

Section 2

Introduction to Hyphenation Patterns

Patterns (of hyphenation) that compete with each other [1].

- pattern is a substring with a piece of information about hyphenation between characters: `hy3ph he2n n2at hen5at`
- odd numbers permit, even numbers forbid hyphenation

Patterns (of hyphenation) that compete with each other [1].

- pattern is a substring with a piece of information about hyphenation between characters: `hy3ph he2n n2at hen5at`
- odd numbers permit, even numbers forbid hyphenation
- patterns are as short as possible to be as general as possible (new compound words, etc.)
- pattern compete with each other: instead of one big set of patterns, decomposition into layered sets generated in *levels*
  p_1 hyphenating patterns generated in level 1, p_2 inhibiting patterns—exceptions for p_1),
  p_3 hyphenating patterns to cover what has not been covered by "$p_1 \wedge \neg p_2$"),…

Hyphenation lookup: an instance of dictionary problem

```
    h y p h e n a t i o n
p1            1n a
p1                1t i o n
p2             n2a t
p2                  2i o
p2        h e2n
p3  h y3p h
p4        h e n a4
p5        h e n5a t
    h0y3p0h0e2n5a4t2i0o0n
```

hy-phen-ation $\rightarrow$ 2 6
$\ldots\rightarrow \ldots$
$\ldots\rightarrow \ldots$
key $\rightarrow$ data

The solution to the dictionary problem:
For the key part (the word) to store
the data part (its division)

Hyphenation lookup: an instance of dictionary problem

```
    h y p h e n a t i o n
p1            1n a
p1                1t i o n
p2             n2a t
p2                  2i o
p2          h e2n
p3  h y3p h
p4          h e n a4
p5          h e n5a t
    h0y3p0h0e2n5a4t2i0o0n
```

hy-phen-ation $\rightarrow$ 2 6
…$\rightarrow$ …
…$\rightarrow$ …
key $\rightarrow$ data

The solution to the dictionary problem: For the key part (the word) to store the data part (its division)

Given the already hyphenated word list of a language (dictionary), *how to generate the patterns*? Liang's task was: less than 5,000 patterns, less than 30,000 bytes per language in format file (RAM during TEX run).

hyphen.tex generation by patgen (Liang, 1983) [1]

level	parameters	patterns	good	bad	good	bad
1	1 2 20 (4)	458	67,604	14,156	76.6%	16.0%
2	2 1 8 (4)	509	7,407	11,942	68.2%	2.5%
3	1 4 7 (5)	985	13,198	551	83.2%	3.1%
4	3 2 1 (6)	1647	1,010	2,730	82.0%	0.0%
5	1 ∞ 4 (8)	1320	6,428	0	**89.3%**	0.0%

A total of 4,919 patterns were obtained in hyphen.tex (27,860 bytes) from Webster's Pocket dictionary (30,000+ words only). *Suffix-compressed packed trie* occupying 5,943 locations, with 181 outputs (less than 1% of original word list).
Patterns find 89.3% of the hyphens in the dictionary. 109 passes through the dictionary are needed.
Generation required about 1 hour of CPU time on PDP-11.

tex-hyphen [3]

- `https://hyphenation.org` is the canonical source of hyphenation patterns for most software
  - TEX
  - web browsers
  - LibreOffice
  - Android (Kindle too!), …

Section 3

Approach

[haɪfəˈneɪʃən˩]

- quality of patterns inconsistent across Slavic languages
- pronunciation, on which syllabic hyphenation is based, is quite similar
- patterns for some languages are really good
- *we can do better*

Pronunciation similar, orthography different

- Пра-га
- Pra-ha
- Pra-ga

International Phonetic Alphabet

ˌɪntərˈnæʃənəl fəˈnɛtɪk ˈælfəˌbɛt

Anti-goals

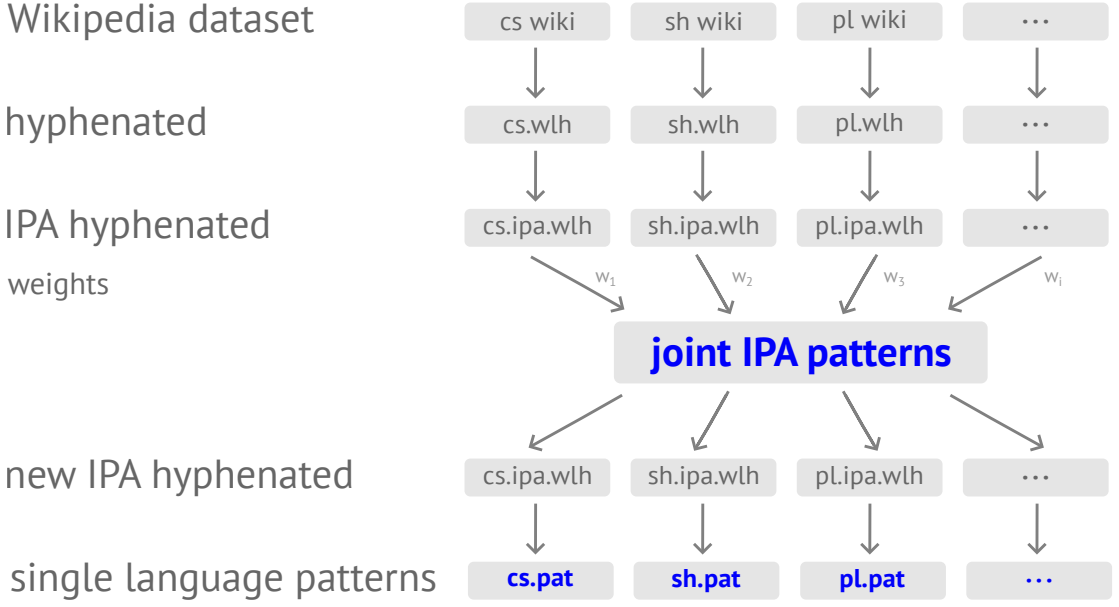
- exert my opinions as a *non-native speaker* into the resulting patterns as I'm not qualified for it
- improve already good patterns

Goals

- improve patterns for languages with no or subpar current patterns with transfer learning
- to develop and deploy the methodology pattern development through transfer learning for several languages in one language family

Section 4

Methodology


Wikipedia dataset
hyphenated
IPA hyphenated
weights
new IPA hyphenated
single language patterns
cs wiki
sh wiki
pl wiki
cs.wlh
sh.wlh
pl.wlh
cs.ipa.wlh
sh.ipa.wlh
pl.ipa.wlh
w1
w2
w3
wi
joint IPA patterns
cs.ipa.wlh
sh.ipa.wlh
pl.ipa.wlh
cs.pat
sh.pat
pl.pat

Source wordlists

Wikipedia dataset

- afaik, hard to acquire clean single-language wordlists
- previously (for Czech and Slovak) provided by Lexical Computing, now unwilling
- reproducibility is important
- $\Rightarrow$ wikipedia
  - cleaned
  - colloquial terms not represented

Transfer of hyphens to IPA

- `espeak-ng` [2] used for generation of IPA
  - consistent across 127 languages
- transfer not trivial!

Transfer of hyphens

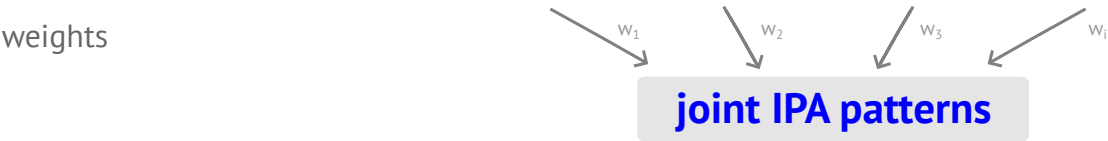
- task: shro - maž - ďo - va - cí + shrˈomaʒɬˌovatsiː ⇒ shrˈo - maʒ - ɬˌo - va - tsiː
- IPA depends on surrounding characters
- where do we put the hyphens?

Transfer of hyphens

```
GCATGCG
GATTACA
---
GCAT GCG
G ATTACA
```

Needleman-Wunsch, algorithm for global alignment

Generation of joint IPA patterns



- *weights* of IPA-hyphenated wordlists crucial to well-performing final patterns
- optimized according to *ground truth* source hyphenation data
- patterns can learn IPA well: `good 99.81 %, bad 0.28 %, missing 0.19 %`
  - challenge is not to overfit; they can infer the language and reproduce original errors
  - won't fix the out-of-distribution samples; anti-goal

Source hyphenated wordlist data

- need ground truth to optimize weights
- need ground truth to validate (separate from optimization of weights!)
  - will probably use native speakers (preferably linguists) for this
  - very few language institutes provide hyphenated words
  - few dictionaries provide hyphenation
- severe lack of definitively-correctly hyphenated words

do you know a good source of hyphenated words for *your* language?

Generation of joint IPA patterns

weights

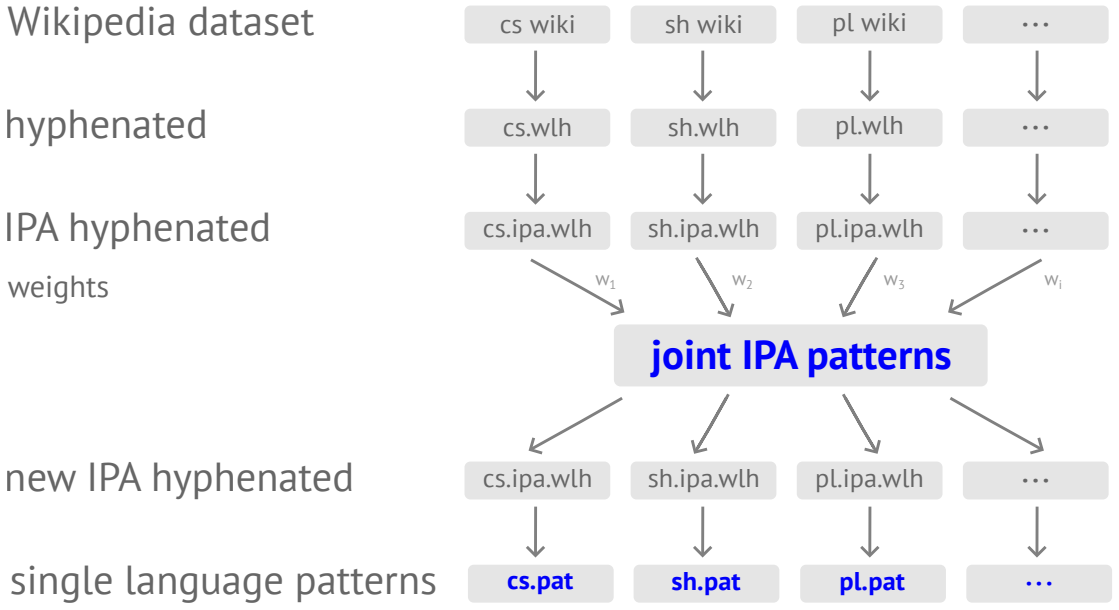
- *weights* of IPA-hyphenated wordlists crucial to well-performing final patterns
- optimized according to *ground truth* source hyphenation data
- to avoid gridsearch in parameter (weight) space, train surrogate model and sample weights to evaluate

Transfer of hyphens from IPA to original

- approach similar to transfer from original to IPA

Final single-language patterns

- easy to generate
- hard to evaluate
- in the absence of reliable ground truth:
  - at least two native speakers hyphenate words, where they match, hyphenation considered good enough
  - compute probability of improvement with new patterns, if $p > 0.95$, propose for inclusion into `tex-hyphen` [3]


Wikipedia dataset
hyphenated
IPA hyphenated
weights
new IPA hyphenated
single language patterns
cs wiki
sh wiki
pl wiki
cs.wlh
sh.wlh
pl.wlh
cs.ipa.wlh
sh.ipa.wlh
pl.ipa.wlh
w1
w2
w3
wi
joint IPA patterns
cs.ipa.wlh
sh.ipa.wlh
pl.ipa.wlh
cs.pat
sh.pat
pl.pat

Section 5

Conclusion

Results

- on a validation wordlist size 15714, which one is best?
  1. `13106 good, 4609 bad, 26574 missed`
  2. `19394 good, 7745 bad, 20286 missed`
  3. `15091 good, 4951 bad, 24589 missed`
  4. `25210 good, 13154 bad, 14470 missed`

Results

- on a validation wordlist size 15714, which one is best?
  1. `13106 good, 4609 bad, 26574 missed`
  2. `19394 good, 7745 bad, 20286 missed`
  3. `15091 good, 4951 bad, 24589 missed`
  4. `25210 good, 13154 bad, 14470 missed`
- shuffled:
  - current Ukrainian patterns
  - transfer from 100 % Slovak
  - transfer from 100 % Ukrainian
  - transfer from 100 % Russian

Results

- on a validation wordlist size 15714, which one is best?
  1. `13106 good, 4609 bad, 26574 missed`
  2. `19394 good, 7745 bad, 20286 missed`
  3. `15091 good, 4951 bad, 24589 missed`
  4. `25210 good, 13154 bad, 14470 missed`
  5. `19308 good, 7620 bad, 20372 missed`

  1. transfer from 100 % Russian
  2. transfer from 100 % Ukrainian
  3. transfer from 100 % Slovak
  4. current Ukrainian patterns
  5. approx 1:1 sk:uk mix

Results

- reason to believe that just through transfer, we can improve the patterns!
  - arguably the garbage in, garbage out approach because those are terrible results
- so we *can* transfer, but we would ideally like to get something in between the original and transferred for better coverage
- obviously we can gridsearch various weight combinations, but can we be smarter about it?

More than weights to tweak!

- `18183 good, 7857 bad, 21497 missed` – german 8 levels parameters
- `10276 good, 3514 bad, 29404 missed` - custom correctoptimized
- `12595 good, 3850 bad, 27085 missed` – custom sizeoptimized

Results

- it is feasible to significantly improve at least current Polish, Croatian, Serbian, and Ukrainian patterns
  - applicable to other language families
- reproducible workflow released [4]

Section 6

Bibliography

Bibliography I

[1] Franklin M. Liang. “Word Hy-phen-a-tion by Com-put-er.” PhD thesis. Stanford University, Aug. 1983, p. 44. URL: https://tug.org/docs/liang/liang-thesis.pdf.

[2] Jonathan Reynolds. *eSpeak NG*. Version 1.50. 2016. URL: https://github.com/espeak-ng/espeak-ng.

[3] Arthur Rosendahl and Mojca Miklavec. *TEX hyphenation patterns*. eng. Accessed 2024-07-16. 2023. URL: http://hyphenation.org/tex.

[4] Ondřej Sojka and Petr Sojka. *patterns workflow repository*. eng. URL: https://github.com/tensojka/patterns.