Expanding Hyphenation Patterns Across Slavic Languages

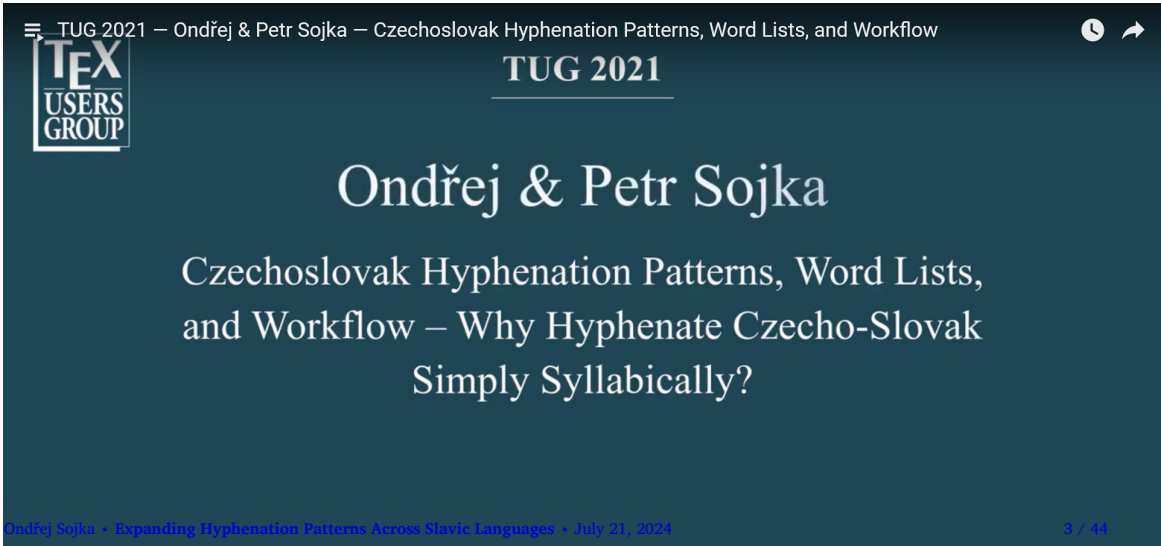
Ondřej Sojka

Faculty of Informatics, Masaryk University

July 21, 2024


EXIT

TUG 2021 – one set of patterns for 2 languages: Czech and Slovak



TUG 2023 Bonn – joint patterns for nine languages

TUG 2024 Prague – transfer learning and sharing of universal syllabification phonetic patterns across Slavic languages

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

Contents

Section 1

Introduction to Hyphenation Patterns

Patterns? Patterns!

> "**pattern** ORIGIN Middle English patron 'something serving as a model', from Old French. The change in a sense is from the idea of *patron giving an example to be copied.* Metathesis in the second syllable occurred in the 16th cent. By 1700 patron ceased to be used of things, and the two forms became differentiated in sense."
> —*New Oxford Dictionary of English, 1998 edition*

One can see the patterns everywhere: rhythm patterns in music or poetry conveying a message, patterns of behavior, letter patterns, …, you name it: *hyphenation patterns*.

Instead of storing the whole ever-growing dictionary of words with hyphenation points, a smaller set of rules called hyphenation patterns are generated from the dictionary, covering most if not all, information from there.

Patterns (of hyphenation) that compete with each other [5].

- 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
$\ldots\rightarrow$ …
$\ldots\rightarrow$ …
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) [5]

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.

`patgen` program: machine learning from data

One of the very first approaches that harnessed the power of data: Liang's program `patgen` for generation of hyphenation patterns from a word list:

- efficient lossy or lossless *compression* of hyphenated dictionary with several orders of magnitude compression ratio.
- generated patterns have minimal length, e.g., shortest context possible, which results in their *generalization* properties.
- hyphenation of out-of-vocabulary words, too.

For Czech, *exact lossless* pattern generation *is feasible* [12] (TUG 2019), while reaching *100% coverage and simultaneously no errors*, and the same holds for 2 language patterns (Czech + Slovak, TUG 2021) [11], 9 languages [10].
Strict pattern minimality (size) is not an issue nowadays.

tex-hyphen [8]

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

Section 2

Hyphenation as Syllabification

Approaches to hyphenation

etymology-based The rule is to cut a word on the border of a compound word or the border of the stem and an affix, prefix, or negation. A typical example is the British hyphenation rules by the Oxford University Press [6].

phonology-based Hyphenation respects the pronunciation of syllables and allows for much more fluent reading. American publishers [2] and the Chicago Manual of Style [1] users prefer this pragmatic approach.

Clear trend towards phonology-based, e.g. syllabic hyphenation.

Rules and examples of syllabic segmentations

- primarily syllabic according to pronunciation
- morphology only secondary (compound words)
- language per se, its vocabulary and hyphenation rules develop in time:
  roz-um (Haller, 1956 [3], prefix roz, stem um) →
  ro-zum (2021, [14] just syllables, etymology forgotten)[1]

[1] Similar shift is in other languages and cultures (UK→US)

Why syllabic patterns?

"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

Why syllabic patterns?

- Proof of concept for **universal** hyphenation patterns presented at RASLAN 2019 workshop [13].
- Superb results for Czech + Slovak patterns [11], consistent markup of syllables increased the quality of joint patterns.
- No hyphenation collisions — almost no words that have different hyphenations in different languages with syllabic hyphenation preferences.
- Available data resources to join the ‘syllabic hyphenation club’.

Section 3

Approach

[haɪfəˈneɪʃənˌ]

- 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

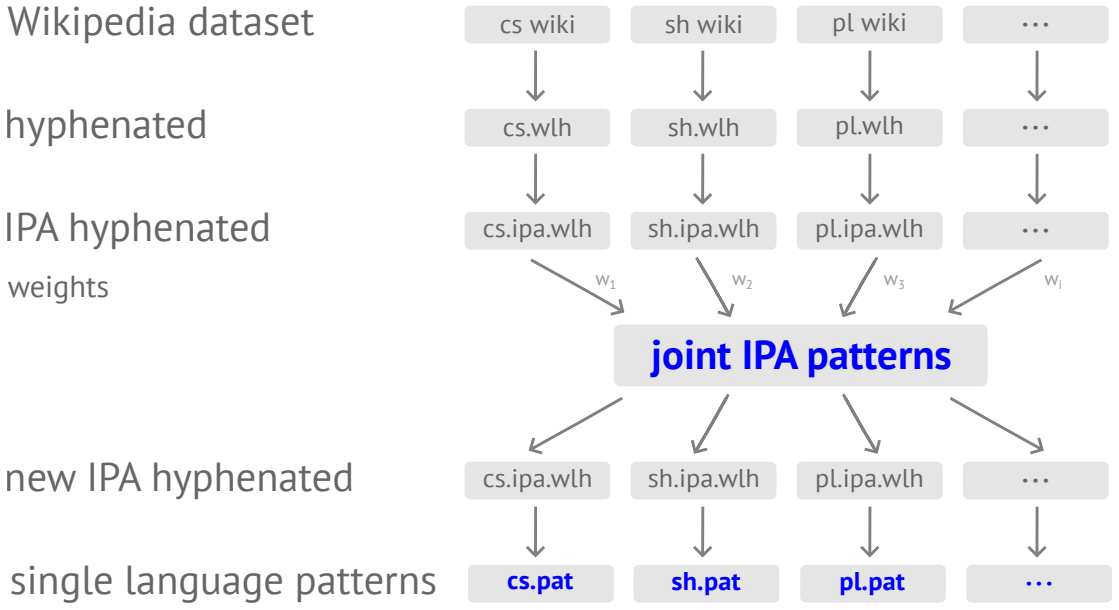
- exert my opinions as a *non-native speaker* into the resulting patterns as I'm not qualified for it
- improve already good patterns

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

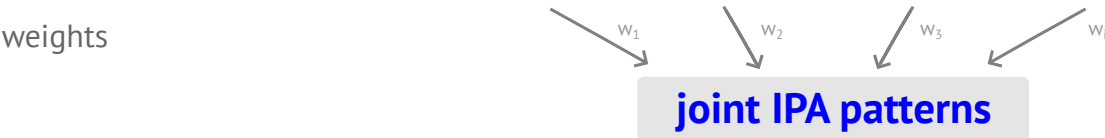
Transfer of hyphens to IPA

- `espeak-ng` [7] used for generation of IPA
  - consistent across 127 languages
- trasnfer not trivial!

Transfer of hyphens

- task: shro - maž - ďo - va - cí + shrˈomaʒɟˌovatsi: ⇒ shrˈo - maʒ - ɟˌo - va - tsi:
- IPA depends on surrounding characters
- where do we put the hyphens?
- $\binom{15}{4} = 1365$ possibilities
- compute Jaro [4] like similarity score
  - $O(n * m)$ where n = length of word, m = number of hyphens
- if there is a tie, compute Levenshtein similarity
  - $O(n * l)$, where n = length of word, l = length of original word
- if there is a tie, convert each 'syllable' to IPA, compute Jaro + Levenshtein

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

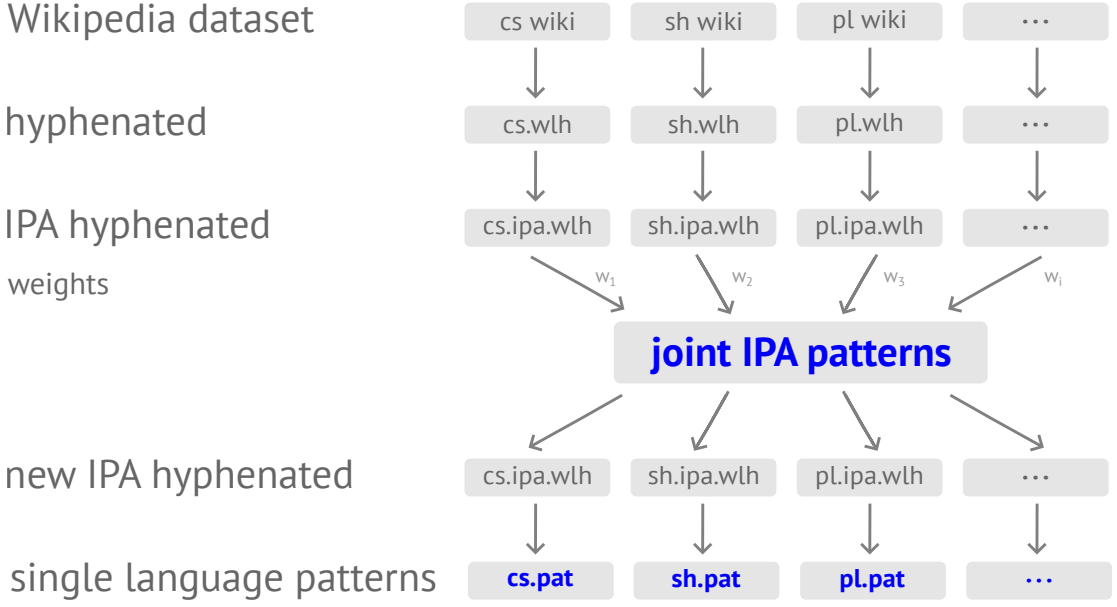
- *weights* of IPA-hyphenated wordlists crucial to well-performing final patterns
- optimized according to *ground truth* source hyphenation data
- to avoid gridsearch in parameter (weight) space, train surrogate model and sample weights to evaluate

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` [8]


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

- it is feasible to significantly improve at least current Polish, Croatian, Serbian, and Ukrainian patterns
  - applicable to other language families
- reproducible workflow released [9]
- resulting single-language patterns will be submitted to `tex-hyphen` by end of year

Future work

- ground truth and robust evaluation setup
- significantly improve at least current Polish, Croatian, Serbian, and Ukrainian patterns
- Slavic universal patterns preparation
- offering new patterns to `tex-hyphen`

That’s it, folks!

If you know a source of ground truth hyphenation data for your Slavic language, reach out!

Questions?

Section 6

Bibliography

Bibliography I

[1] Anonymous. *The Chicago Manual of Style*. 17th ed. Chicago: University of Chicago Press, Sept. 2017, p. 1146. ISBN: 9780226287058.

[2] Philip Babcock Gove and Merriam Webster. *Webster's Third New International Dictionary of the English language Unabridged*. Springfield, Massachusetts, U.S.A: Merriam-Webster Inc., Jan. 2002.

[3] Jiří Haller. *Jak se dělí slova (How the Words Get Hyphenated)*. Státní pedagogické nakladatelství Praha, 1956.

[4] M. A. Jaro. "Probabilistic linkage of large public health data file." In: *Statistics in Medicine* 14.5–7 (1995), pp. 491–498. DOI: 10.1002/sim.4780140510.

[5] 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.

Bibliography II

[6] R. E. Allen, ed. *The Oxford Spelling Dictionary*. Vol. II. The Oxford Library of English Usage. Oxford University Press, 1990, p. 299.

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

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

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

[10] Ondřej Sojka, Petr Sojka, and Jakub Máca. “A roadmap for universal syllabic segmentation.” eng. In: *TUGboat* 44.2 (2023). ISSN: 0896-3207. URL: https://doi.org/10.47397/tb/44-2/tb137sojka-syllabic.

Bibliography III

[11] Petr Sojka and Ondřej Sojka. “New Czechoslovak Hyphenation Patterns, Word Lists, and Workflow.” eng. In: *TUGboat* 42.2 (2021). ISSN: 0896-3207. URL: https://doi.org/10.47397/tb/42-2/tb131sojka-czech.

[12] Petr Sojka and Ondřej Sojka. “The Unreasonable Effectiveness of Pattern Generation.” In: *TUGboat* 40.2 (2019), pp. 187–193. URL: https://tug.org/TUGboat/tb40-2/tb125sojka-patgen.pdf.

Bibliography IV

[13] Petr Sojka and Ondřej Sojka. “Towards Universal Hyphenation Patterns.” In: *Proceedings of Recent Advances in Slavonic Natural Language Processing—RASLAN 2019*. Ed. by Aleš Horák, Pavel Rychlý, and Adam Rambousek. https://is.muni.cz/publication/1585259/?lang=en. Karlova Studánka, Czech Republic: Tribun EU, 2019, 63–68. URL: https://nlp.fi.muni.cz/raslan/2019/paper13-sojka.pdf.

[14] *Internetová jazyková příručka (Internet Language Reference Book)*. Czech. URL: https://prirucka.ujc.cas.cz/?id=135 (visited on 07/18/2019).