MUNI
FI

New Czechoslovak Hyphenation Patterns, Word Lists, and Workflow

Why Hyphenate Czecho-Slovak?

Petr Sojka, Ondřej Sojka

Faculty of Informatics, Masaryk University

August 8, 2021


EXIT
MAKE TEX GREAT AGAIN
REGISTERED TRADEMARK
DENIM BRAND
SHERATON
WEB 10
EPSON

Contents

Section 1

Introduction to Hyphenation Patterns

Patterns (in general)

> “**pattern** ORIGIN Middle English patron ‘something serving as a model’, from Old French. The change in 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*

Patterns everywhere: rhythm patterns in music or poetry conveying message, patterns of behaviour, letter patterns, …, you name it: *hyphenation patterns*.

Patterns (of hyphenation) that compete each other

Frank Liang, DEK's student at Stanford (Ph.D., 1983), developed *the method* and algorithms for hyphenation based on the idea of competing patterns of varying length to cope with exceptions. [3].

- general, language-independent method
- pattern is a substring with a information about hyphenation between characters: `hy3ph he2n n2at hen5at .euro7`
- 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 each other: instead of one big set of patterns, decomposition into several layered generated sets: *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

Solution to the dictionary problem: For 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) [3]

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 (4,447 unique) obtained in hyphen.tex (27,860 bytes) from Webster 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.

hyphen.tex used as a default for \language=0 in every TEX installation

```
% The Plain TeX hyphenation tables
% [NOT TO BE CHANGED IN ANY WAY!]
% Unlimited copying and redistribution of this file
% are permitted as long as this file is not modified.
% Modifications are permitted, but only if
% the resulting file is not named hyphen.tex.
\patterns{% just type <return> if you're not using INITEX
.ach4 .ad4der .af1t .al3t .am5at
...
```

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.

Generally, *exact lossless* pattern *minimization is non-polynomial* by reduction to the minimum set cover problem [5]. For Czech, *exact lossless* pattern generation *is feasible* [7] (TUG 2019), while reaching *100% coverage and simultaneously no errors*.
Strict pattern minimality (size) is not an issue nowadays.

csskhyphen.pat

```
% Czechoslovak hyphenation patterns
% Generated with correct optimized parameters
%   (cs-sojka-correctoptimized.par)
% For further information, see TUGboat issue
%   with proceedings of TUG 2021
% For source code, see github.com/tensojka/cshyphen
% MIT License
% ...
.ad3aw .ads4 .af3r .ai4č .ak3ry .al3s .as3k .as3t
.at3at. .bel3h
...
```

"An important feature of a learning machine is that its teacher will often be very largely ignorant of quite what is going on inside, although he may still be able to some extent to predict his pupil's behaviour." — *Alan Turing, Mind 59:433–460, 1950*

Section 2

Czech, Slovak and Czechoslovak

Why hyphenate Czecho-Slovak?

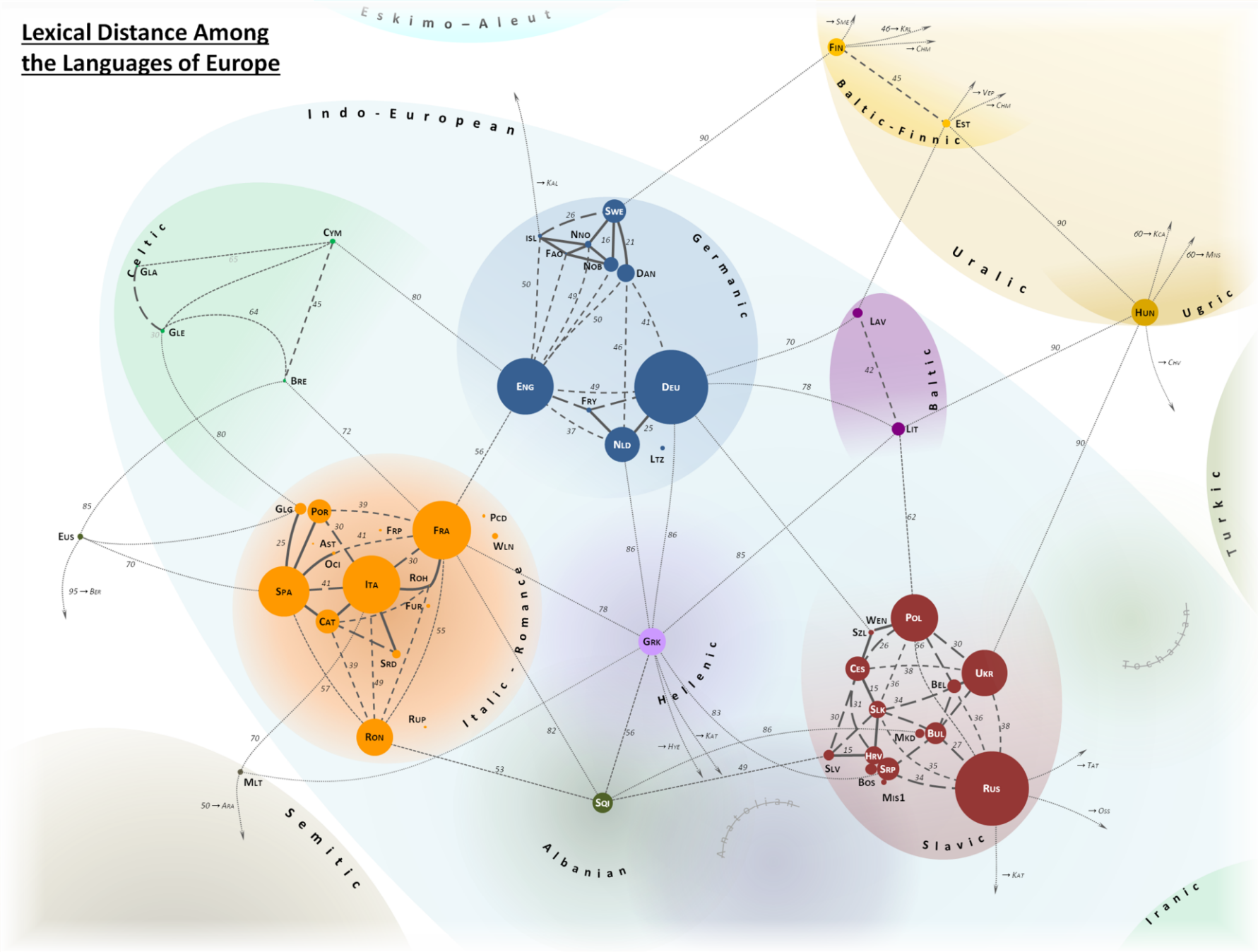
- separate, but very similar languages
- very practical
- 40+% of students at Faculty of Informatics, MU in Brno, Czech Republic, are Slovak

- proof of concept for **universal** hyphenation patterns presented at RASLAN 2019 workshop [8]

Is there no word that has different hyphenations in the covered languages?

Then we can cover multiple languages with one set of hyphenation patterns.

Lexical Distance Among the Languages of Europe



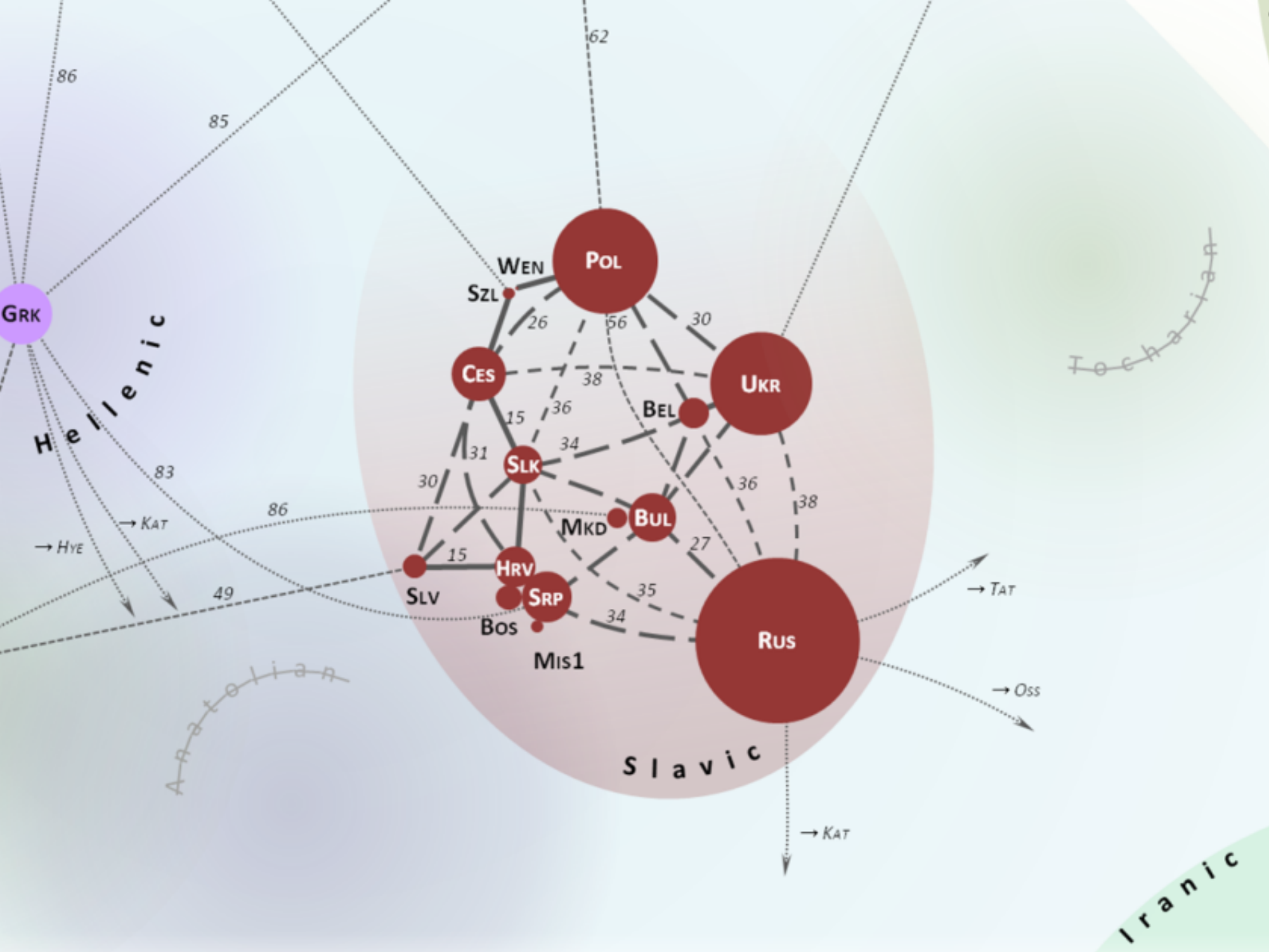
Abbreviations

Albanian		**Semitic**		**Celtic**		**Germanic**				**Italic-Romance**				**Slavic**					
sqi	Albanian	mlt	Maltese	cym	Welsh	deu	German	isl	Icelandic	ita	Italian	oci	Occitan	rus	Russian	bos	Bosnian		
Baltic		**Hellenic**		gle	Irish	eng	English	fao	Faroese	fra	French	fur	Friulian	pol	Polish	slv	Slovene		
lit	Lithuanian	grk	Greek	bre	Breton	nld	Dutch			spa	Spanish	pcd	Picard	ukr	Ukrainian	mkd	Macedonian		
lav	Latvian			gla	Scottish Gaelic	swe	Swedish			ron	Romanian	frp	Franco-Provençal	ces	Czech	szl	Silesian		
				Uralic		dan	Danish			cat	Catalan	rup	Aromanian	srp	Serbian	mis1	Montenegrin		
						nob	Norwegian Bokmål			por	Portuguese	ast	Asturian	bul	Bulgarian	wen	Sorbian		

Lexical Distance: 25 35 50 70

Speakers: 100mil, 50mil, 25mil

alternativetransport.wordpress.com
by
S. Steinbach 2015


62
86
85
83
86
49
Hellenic
GRK
→ HYE
→ KAT
WEN
SZL
POL
26
56
30
CES
38
UKR
15
36
BEL
31
34
SLK
30
36
38
MKD
BUL
27
15
HRV
SLV
SRP
35
34
BOS
MIS1
RUS
→ TAT
→ OSS
→ KAT
Slavic
Anatolian
Tocharian
Iranic

Two approaches to hyphenation

- etymology-based
- phonology-based

Foreign words

- hyphenation must not disrupt reading
- English pronunciation: goo-gle
- pronunciation: go-o-g-le

Hyphenation in Czech

Rules are published at [2]:
`https://prirucka.ujc.cas.cz/?id=135`

- 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, prefix roz, stem um) $\rightarrow$
  ro-zum (2021, just syllables, etymology forgotten)[1]

[1] Similar shift is in other languages and cultures (UK$\rightarrow$US)

Hyphenation in Slovak

Rules are published at [4] `https://www.juls.savba.sk/ediela/psp2000/psp.pdf#page=25`

- morphology primary according to the Ľ. Štúr Institute of Linguistics
- syllabic hyphenation secondary
- morphological boundaries are often also syllabic boundaries
- current patterns hyphenate mostly syllabically
- pupils learn to hyphenate syllabically

Observations

- telling apart the prefix *nej-* from *ne-* is a problem
- Slovak patterns hyphenate *syllabically* most of the time, contrary to recommendations of the Slovak language institute.
  - ne-na_u-čí
- is this correct behavior?
  - vy-ma-lo-va-ných
  - vy-maľ-o-va-ných

Tradeoffs

- impossible to hyphenate both languages “by the book”
- our patterns hyphenate syllabically
  - allows for easier reading
  - easier to hyphenate
  - lazier approach wins with language users

Section 3

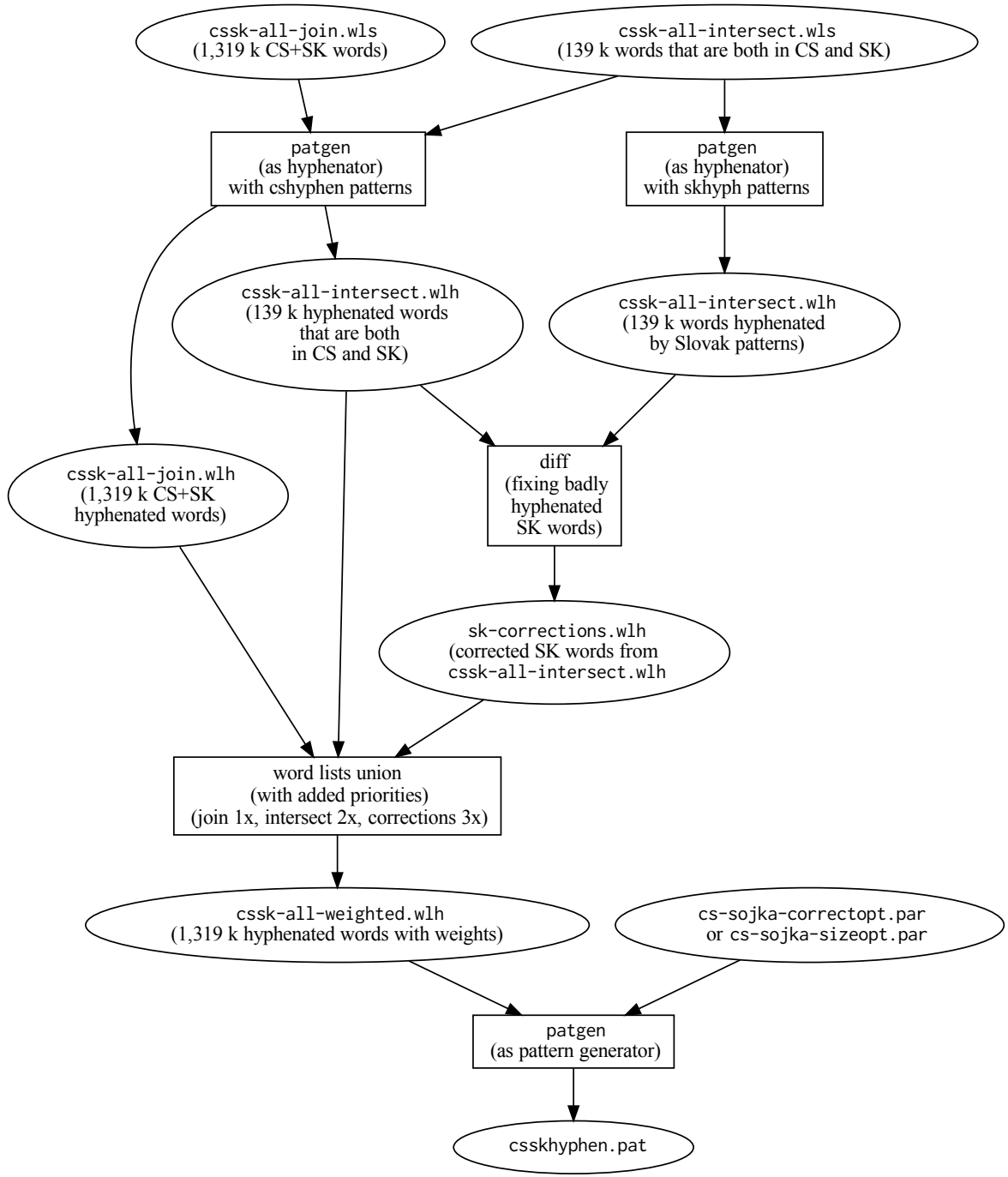
Pattern Generation

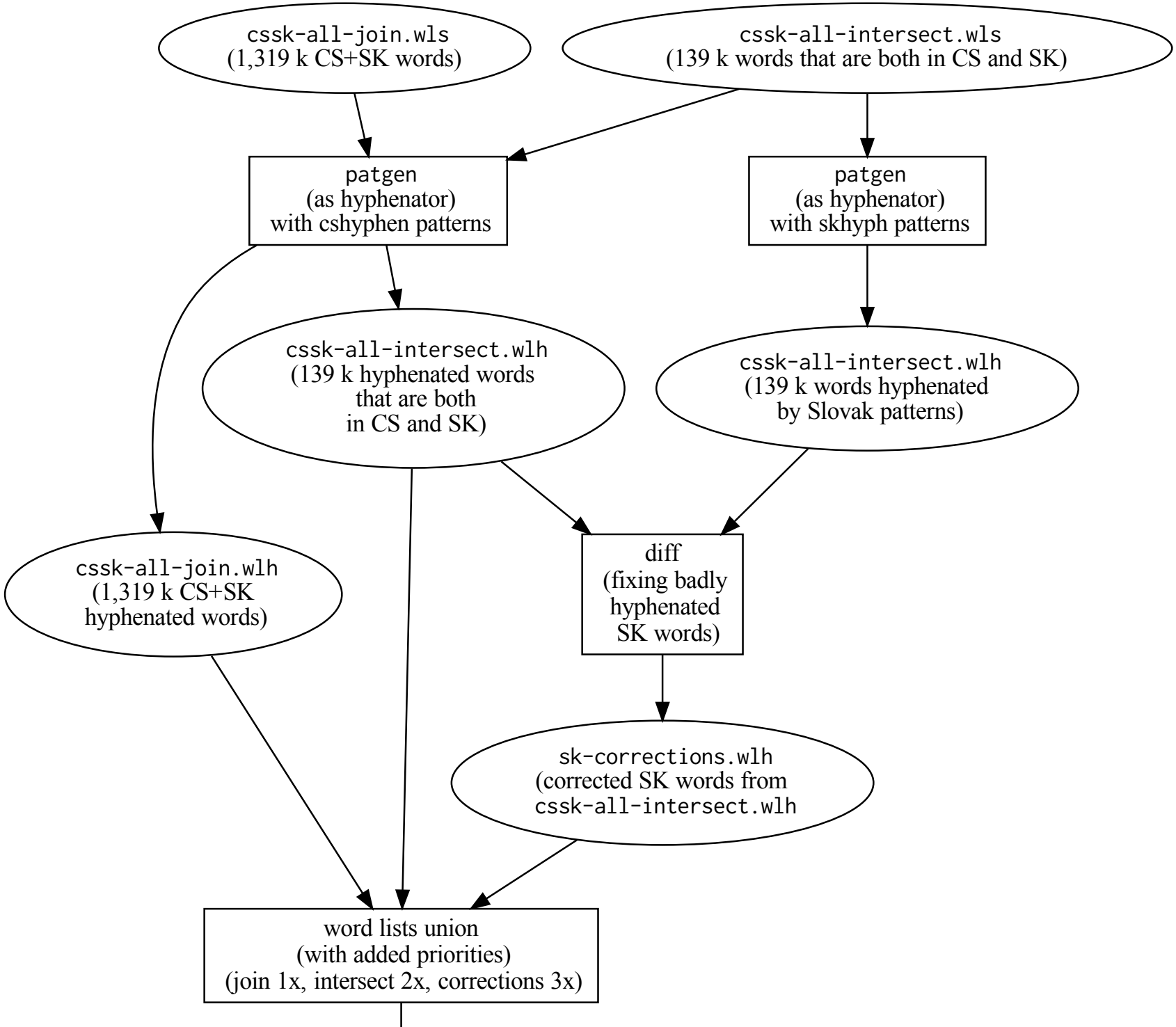
We have

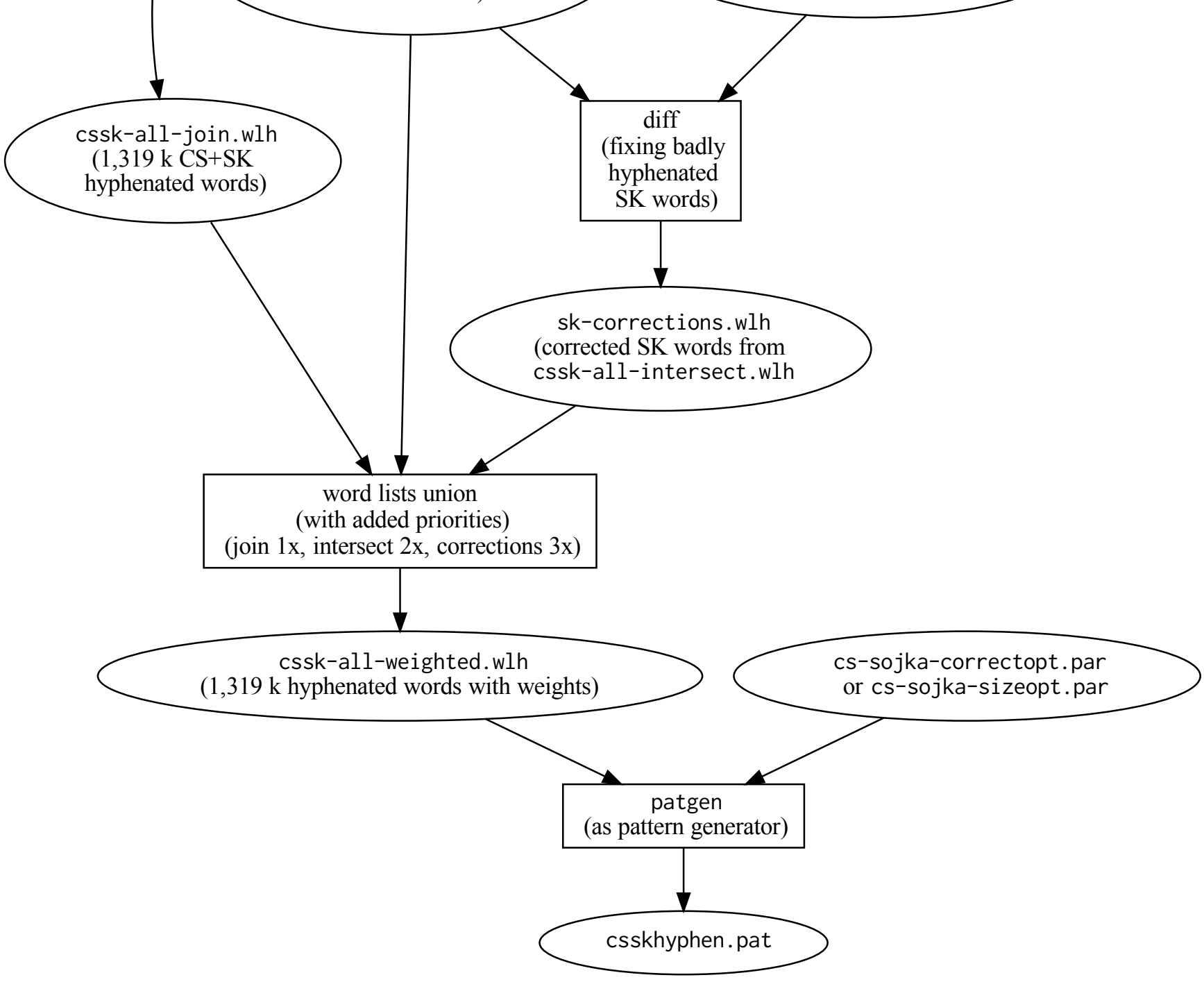
- good Czech patterns
- mediocre Slovak patterns (hand made, not generated) [1]
- Czech and Slovak word lists crawled and edited [8]
- Czech word list hyphenated with Czech patterns
- human hyphenators

We want a workflow, leading to

Patterns that *miss* very few or *no* hyphenation points and make *no mistakes* in either language


cssk-all-join.wls
(1,319 k CS+SK words)
cssk-all-intersect.wls
(139 k words that are both in CS and SK)
patgen
(as hyphenator)
with cshyphen patterns
patgen
(as hyphenator)
with skhyph patterns
cssk-all-intersect.wlh
(139 k hyphenated words
that are both
in CS and SK)
cssk-all-intersect.wlh
(139 k words hyphenated
by Slovak patterns)
cssk-all-join.wlh
(1,319 k CS+SK
hyphenated words)
diff
(fixing badly
hyphenated
SK words)
sk-corrections.wlh
(corrected SK words from
cssk-all-intersect.wlh
word lists union
(with added priorities)
(join 1x, intersect 2x, corrections 3x)
cssk-all-weighted.wlh
(1,319 k hyphenated words with weights)
cs-sojka-correctopt.par
or cs-sojka-sizeopt.par
patgen
(as pattern generator)
csskhyphen.pat


cssk-all-join.wls
(1,319 k CS+SK words)
cssk-all-intersect.wls
(139 k words that are both in CS and SK)
patgen
(as hyphenator)
with cshyphen patterns
patgen
(as hyphenator)
with skhyph patterns
cssk-all-intersect.wlh
(139 k hyphenated words
that are both
in CS and SK)
cssk-all-intersect.wlh
(139 k words hyphenated
by Slovak patterns)
cssk-all-join.wlh
(1,319 k CS+SK
hyphenated words)
diff
(fixing badly
hyphenated
SK words)
sk-corrections.wlh
(corrected SK words from
cssk-all-intersect.wlh
word lists union
(with added priorities)
(join 1x, intersect 2x, corrections 3x)


cssk-all-join.wlh
(1,319 k CS+SK
hyphenated words)
diff
(fixing badly
hyphenated
SK words)
sk-corrections.wlh
(corrected SK words from
cssk-all-intersect.wlh
word lists union
(with added priorities)
(join 1x, intersect 2x, corrections 3x)
cssk-all-weighted.wlh
(1,319 k hyphenated words with weights)
cs-sojka-correctopt.par
or cs-sojka-sizeopt.par
patgen
(as pattern generator)
csskhyphen.pat

Pattern generation – custom parameters

Level	Patterns	Good	Bad	Missed	Lengths	Params
1	830	2,819,833	470,649	35,908	1 3	1 3 12
2	1,590	2,748,581	3,207	107,160	2 4	1 1 5
3	2,766	2,852,334	12,197	3,407	3 6	1 2 4
4	1,285	2,851,931	986	3,810	3 7	1 4 2

Custom parameters are used to filter out still manageable amount of words to check and fix in the primary word lists.

Pattern generation – correct optimized parameters

Level	Patterns	Good	Bad	Missed	Lengths	Params
1	2,032	2,800,136	242,962	55,605	1 3	1 5 1
2	2,009	2,791,326	10,343	64,415	1 3	1 5 1
3	3,704	2,855,554	11,970	187	2 6	1 3 1
4	1,206	2,854,794	33	947	2 7	1 3 1

With correct optimized setting we get no errors (33 words could be added as patterns), with only 4 levels used.

10-fold cross validation results

Parameters	Good	Bad	Missed
correctopt	99.81%	0.15%	0.04%
custom	99.64%	0.22%	0.14%
sizeopt	99.41%	0.18%	0.40%

Generalization abilities are checked by 10-fold cross validation: patterns are generated from 9/10 of word list, and performance is measured on yet unseen "new" words. This is repeated 10 times and the performance averaged.

Comparison – training (not validation) results

Word list	Parameters	Good	Bad	Missed	Size
Slovak	[1, by hand]	N/A	N/A	N/A	20 kB
Czech	correctopt [7]	99.76%	2.94%	0.24%	30 kB
Czech	sizeopt [7]	98.95%	2.80%	1.05%	19 kB
Slovak	[6, Table 1]	99.94%	0.01%	0.06%	56 kB
Czechoslovak	sizeopt	99.67%	0.00%	0.33%	40 kB
Czechoslovak	correctopt	99.99%	0.00%	0.01%	45 kB
Czechoslovak	custom	99.87%	0.03%	0.13%	32 kB

Section 4

Conclusions

Summary of outcomes

1. new Czechoslovak patterns
2. new Czech and Slovak word lists
3. new workflow for hyphenated word list acquisition
4. new workflow for pattern development and customized pattern generation

All primary sources and generated patterns file csskhyphen.pat are posted in public github repository `https://github.com/tensojka/cshyphen`. Documentation is to be found in the paper in the TUG 2021 proceedings.

Next steps

- inclusion to `tex-hyph` package
- two possible approaches:
  - to introduce new ‘Czecho-Slovak language’ in babel, polyglossia, or
  - new patterns to be used just as replacements for both old Czech and Slovak patterns?

That's it, folks! Comments, suggestions, pull requests are welcome!

Questions?

Section 5

Bibliography

Bibliography I

[1] Jana Chlebíková. “Ako rozdeliť (slovo) Československo (How to hyphenate the word Czechoslovakia)”. In: *Zpravodaj $\mathcal{CS}$TUG* 1.4 (Apr. 1991), 10–13. DOI: 10.5300/1991-4/10. URL: https://cstug.cz/bulletin/pdf/bul914.pdf#page=12.

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

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

Bibliography II

[4] Ľ. Štúr Institute of Linguistics of the Slovak Academy of Sciences (SAS), ed. *Pravidlá slovenského pravopisu (Rules of Slovak Grammar)*. Slovak. Third, updated printing. Bratislava: Veda, publisher of SAS, 2000. URL: https://www.juls.savba.sk/ediela/psp2000/psp.pdf (visited on 08/07/2021).

[5] Petr Sojka. “Competing Patterns in Language Engineering and Computer Typesetting”. PhD thesis. Faculty of Informatics, Jan. 2005, pp. xiii+140.

[6] Petr Sojka. “Slovenské vzory dělení: čas pro změnu?” In: *Proceedings of SLT 2004, 4th seminar on Linux and TEX*. Znojmo: Konvoj, 2004, 67–72. URL: https://fi.muni.cz/usr/sojka/papers/skhyp.pdf.

Bibliography III

[7] 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.

[8] 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, pp. 63–68. URL: https://nlp.fi.muni.cz/raslan/2019/paper13-sojka.pdf.