New Czechoslovak hyphenation patterns, word lists, and workflow*

Petr Sojka, Ondřej Sojka

Abstract

Space- and time-effective segmentation and hyphenation of natural languages remain at the core of every document preparation system, web browser, or mobile rendering system. We use the unreasonable effectiveness of pattern generation with `patgen`. It is possible to use hyphenation patterns to solve the dictionary problem also for closely related languages, without compromise. In this article, we show how we applied the marvelous effectiveness of `patgen` for the generation of the new Czechoslovak hyphenation patterns that cover both Czech and Slovak languages.

We show that developing universal, up-to-date, high-coverage and highly generalized hyphenation patterns is feasible, generated from semi-automatically prepared word lists from actual language usage. We evaluate the new approach and argue that the new Czechoslovak hyphenation patterns bring significant coverage and generalization improvements, and space savings. We share all the data, word lists, and workflow for reproducibility and usage.

> "Any respectable word processing package includes a hyphenation facility. Those based on an algorithm, also called logic systems, often break words incorrectly." Major Keary in [11]

1 Introduction

Space- and time-effective segmentation and hyphenation of natural languages remain at the core of every document preparation system, be it TEX, a modern web browser, or mobile rendering system.

The Unicode Standard supports 5,000 languages that are still in use today. Each of these languages is on the move. A digital typographic system that supports Unicode and its languages in full should support hyphenation in the form of algorithms, rules, or patterns.

However, languages are "moving targets". Vocabulary changes (e.g., a language adopts new words). Meanings of individual words change in time (e.g., gay in English). The importance of word etymology and segmentation changes as well. The word `roz-um` (understanding) hyphenated in 1956 [7] according to prefix `roz`, signaling separation, and suffix `um`, signaling knowledge is now perceived as a single stem `rozum` (intelligence, mind). Thus also word hyphenation algorithms should adapt accordingly from time to time to match language usage.

There are essentially two quite different approaches to hyphenation:

etymology-based The rule is to cut a word on the border of a compound word or the boundary of stem and affix, prefix, or negation. A typical example is the British hyphenation rules from the Oxford University Press [1].

phonology-based Hyphenation follows the pronunciation of syllables, allowing for much more fluent reading. Syllabification is not followed near word borders (in the same languages) — hyphenation is forbidden when close to word borders. American publishers [6] and the *Chicago Manual of Style* [4] users prefer this pragmatic approach.

There is a trade-off between the two: one prefers visual highlighting of the word meaning etymology as British do, or likes phonology — convenient reading across the lines.

There is high diversity among languages, but what is the same is that the meaning is conveyed by syllables of the language [15]. There is also high diversity among languages' spelling, but what is the same is that the mapping from phonology to spelling is almost lossless. And there is high diversity in the language hyphenation rules, but when phonology-based hyphenation is preferred, the syllable definition based on consonant and vowel segments is the same for all languages, providing a chance to develop one universal syllable-based segmentation algorithm.

Czech and Slovak are very close languages. Citizens of Czechoslovakia understood both before the states split in 1993. The syllabification and pronunciation rules are the same. We spotted a clear trend towards phonology-based hyphenation. The differences in spelling are rule-based. These observations led us to the idea of common Czechoslovak hyphenation patterns usable for both languages.

This paper evaluates the feasibility of the development of *universal* phonology-based (syllabic) hyphenation patterns. As a case study, we describe the development of Czechoslovak hyphenation patterns from word lists of Czech [20, 21, 28] and Slovak [23]. We generated new patterns from word lists captured from actual language use during the last decade. We rigorously evaluated new patterns as superior to the current specific Czech and Slovak patterns. We document our reproducible workflow and all resources in a public repository. We conclude by outlining further possible hyphenation pattern developments to meet today's demands.

* This is a significantly updated and enriched version of the paper published in CSTUG's *Zpravodaj* [27].

doi.org/10.47397/tb/42-2/tb131sojka-czech

"Hyphenation does not lend itself to any set of unequivocal rules. Indeed, the many exceptions and disagreements suggest it is all something dreamed up at an anarchists' convention." Major Keary in [11]

2 Syllable segmentation methods

The core idea is to develop shared hyphenation patterns for phonology-based languages. If these languages share pronunciation rules, homographs from different languages typically do not cause problems, as they are hyphenated the same [3, 7, 9, 31]. There are occasional cases where the break in a compound word dictates a hyphenation point contrary to phonology (`roz-um` vs. `ro-zum`). These could be solved by not allowing the hyphenation of this particular word around this specific break.

Marchand et al. [16] showed that data-driven approaches to syllabification algorithms outperform rule-based ones, reaching accuracy levels around 95% per single language. Bartlett et al. [2] developed a machine learning approach for automatic syllabification, motivated by the needs of letter-to-phoneme conversion. Trogkanis et al. [29] used conditional random fields for word hyphenation and compared the accuracy and other metrics with the original technique of Liang [14]. Their results abstracted heuristics to optimize generated patterns by `patgen` [8], diminishing achievable performance by Liang's technique. A recent study on syllabification [13] shows that even in comparison with the latest "deep" neural approaches, fine-tuned `patgen` performance beats them in both accuracy and performance.

Recently, there have been attempts to tackle the word segmentation problem in different languages by Shao et al. [18]. The algorithm is error-prone, but it was developed primarily for speech recognition and language representation tasks. Due to the nonzero error rate, its applicability to the hyphenation task is limited. In a typesetting system, the hyphenation algorithm must cover all exceptions and not tolerate any errors.

We recently showed that the `patgen` approach of pattern generation from word list is unreasonably effective [26]. One can set the parameters of the generation process so that the patterns cover 100% of hyphenation points, and their size remains reasonably tiny. We compressed the word list with 3,000,000 hyphenated words into 30,000 bytes of the packed trie data structure for the Czech language. That means achieving a compression ratio of several orders of magnitude with 100% coverage and nearly zero errors [26]. For a similar language such as Slovak, the pronunciation is very similar, syllable-forming principles are the same, and compositional rules and prefixes are pretty close, if not identical.

We have decided to verify the approach by developing hyphenation patterns that will hyphenate *both* Czech and Slovak words without errors, with only a few missed hyphens. The missed hyphen will appear *only* in words like `oblít` where *meaning* of the term is needed for the decision: `o-blít` or `ob-lít`.

The clear trend, at least in the Czech hyphenation codification books from Haller [7] via [28] used so far in TEX and Word [3], to currently-maintained word lists in [9], reflect gradual movement from etymology to phonology for better syllabic pronunciation when reading hyphenated words. The context-dependent hyphenation decision to resolve such preferences and meaning ambiguities are needed only sporadically.

We needed to create lists of correctly hyphenated Czech and Slovak words to generate these hyphenation patterns.

3 Data preparation

For our work, Lexical Computing CZ donated word lists with frequencies for Czech and Slovak from the TenTen family of corpora [10, 12]. These corpora were drawn from the Internet within the last decade. They contain words used in both languages.

The Czech word list was cleaned up and extended as described by us [24, 25, 26], using the Czech morphological analyzer `majka`. Contrary to the German database, we opted for the simplest format possible, allowing easy enrichment and editing of word lists.

For the generalization of hyphenation rules by `patgen`, we do not need the word list to be as complete as possible, so we used only those words that appeared more than ten times. The final word list file `cs-all-cstenten.wls` contained 606,494 words.

For Slovak, we obtained 1,048,860 Slovak words with a frequency higher than ten from 2011 SkTenTen corpora [10]. We only used words with a frequency higher than thirty comprised of only ISO Latin 2 characters, obtaining a file `sktenten.wls` with 544,609 words.

By joining both language files, we got 967,058 Czech and Slovak words in `cssk-all-join.wls`, of which 106,016 were contained in the intersection of both word lists: `cssk-all-intersect.wls`.

4 Pattern development

Figure 1 illustrates the workflow of the Czechoslovak pattern development. We have used recent, accurate Czech patterns [26] for the hyphenation of the joint Czech and Slovak word list. We had to fix incorrect

hyphenation points manually, typically near the prefix and stem boundary when phoneme-based hyphenation point was one character away from the seam of the prefix or compound word: `neja-traktivnější`, `neja-teistič tější`, `neje-kologič tější`.

We then hyphenated words used in both languages also by the current Slovak patterns. There were only a few word hyphenations that needed to be corrected — we created the file `sk-corrections.wlh` that contained the fixed hyphenated words. Finally, we used them as input to `patgen` with a higher weight during the generation of the final Czechoslovak hyphenated patterns.

We did not pursue 100% coverage at all costs because the source data is noisy, and we do not want the patterns to learn all the typos and inconsistencies. We expand on this in the Jupyter notebook [19]. Gentle readers may also find the scripts used there.

5 Evaluation

We evaluated the quality of developed patterns by two metrics. *Coverage* of hyphenation points in the training word list tells how the patterns correctly predicted hyphenation points used in training. *Generalization* means how the patterns behave on unseen data, on words not available in the data used during `patgen` training.

We see the coverage and generalization as a *classification* task, i.e., how the patterns classify hyphenation points in the training and testing word lists, respectively.

5.1 Classification

To evaluate the classification results, there are four numbers in the contingency matrix that compare hyphenation point prediction by patterns with the ground truth expressed in the wordlist: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In tables 1–4 on page 156, we report:

Good sum or percentage of found hyphenation points (TP),

Bad sum or percentage of badly suggested hyphenation points (FP, type 1 error),

Missed sum or percentage of missed hyphenation points (FN, type 2 error).

Type 1 errors are more severe than type 2 errors in our hyphenation points setup. Nonzero **bad** results do not necessarily mean that the patterns performed poorly. Just the opposite holds — the patterns have found a rule that the ground truth wordlist does not obey. In other words, the inconsistency needs fixing in the underlying word list rather than emitting the pattern for a valid exception. We practiced manual inspection of bad hyphenation points during the development of the word list.

5.2 Generalization

We used tenfold cross-validation to assess the generalization properties, that is, leaving out one-tenth of the training set to evaluate the patterns' effectiveness on unseen words. We show the results in Table 5. The evaluation metrics differ slightly with different `patgen` parameters, with the best results achieved when we maximize the coverage of the training set.

The achieved results show that both evaluation metrics are close to perfection. We can either opt for perfect coverage and reach it or push to maximize generalization qualities and performance on unseen words. In the first case, we achieve essentially lossless compression of wordlist hyphenation points by the developed pattern. In the second, we miss only less than 1% of valid hyphenation points. Achieving that for two languages in parallel seems like a good result. It is feasible to continue merging additional word lists to develop generic patterns for syllabically hyphenated languages.

We do not know pattern performance for most of the other available patterns as there are no word lists to use for the evaluation and comparison.

> "Esoteric Nonsense? Hyphenation is neither anarchy nor the sole province of pedants and pedagogues.... Used in moderation, it can make a printed page more visually pleasing. If used indiscriminately, it can have the opposite effect, either putting the reader off or causing unnecessary distraction. If the intended audience is bound to read the work (a user manual, for example), poor hyphenation practice may not matter. If the author wants to attract and hold an audience, then hyphenation needs just as careful attention as any other aspect of presentation." Major Keary in [11]

6 Conclusion and summary

We have shown that the development of common hyphenation patterns for several languages with similar pronunciations is feasible. `Patgen` was able to generalize hyphenation rules for both languages with only a negligible increase in the size of the generated patterns.

The resulting Czechoslovak patterns hyphenate Czech and Slovak *much better* than the former single-language patterns, with much higher coverage, zero error rate, and evaluated generalization. The whole process is reproducible, is documented, and available as a Jupyter demo notebook with source code [19].

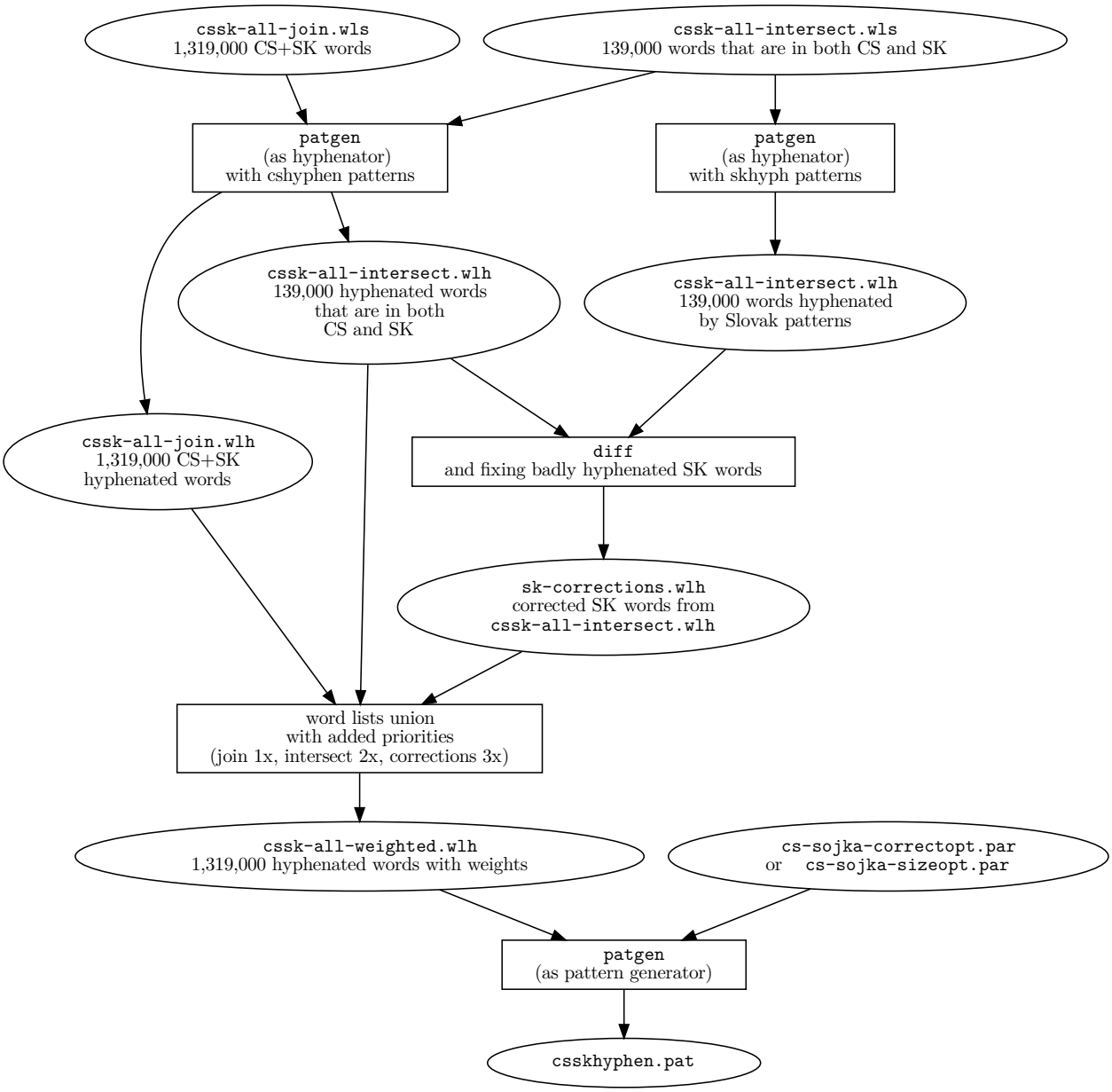

Figure 1: The whole pattern development workflow is showed above from top:
a) Czech and Slovak word lists collection, [26] and intersection;
b) bootstrapping hyphenated word lists with syllabic Czech patterns;
c) checking and fixing by deploying the rarely-used `patgen` weighting for Slovak words common with Czech ones;
d) generation of final patterns.
The whole workflow and scripts are available in the public repository [19].

Table 1: Statistics from the generation of Czechoslovak hyphenation patterns with *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

Table 2: Statistics from the generation of Czechoslovak hyphenation patterns with *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

Table 3: Statistics from the generation of Czechoslovak hyphenation patterns with *size optimized* parameters.

Level	Patterns	Good	Bad	Missed	Lengths	Params
1	419	2,833,402	667,031	22,339	1 3	1 2 20
2	1,506	2,430,120	1,188	425,621	2 4	2 1 8
3	3,579	2,846,112	15,881	9,629	3 5	1 4 7
4	2,401	2,843,657	4	12,084	4 7	3 2 1

Table 4: Comparison of the efficiency of different approaches to hyphenating Czech and Slovak. Note that the Czechoslovak patterns are comparable in size and quality to single-language ones — there is only a negligible difference compared to, e.g., purely Czech patterns.

Word list	Parameters	Good	Bad	Missed	Size	Patterns
Slovak	[5, by hand]	N/A	N/A	N/A	20 kB	2,467
Czech	correctopt [26]	99.76%	2.94%	0.24%	30 kB	5,593
Czech	sizeopt [26]	98.95%	2.80%	1.05%	19 kB	3,816
Slovak	[22, Table 1]	99.94%	0.01%	0.06%	56 kB	2,347
Czechoslovak	sizeopt	99.67%	0.00%	0.33%	40 kB	7,417
Czechoslovak	correctopt	99.99%	0.00%	0.01%	45 kB	8,231
Czechoslovak	custom	99.87%	0.03%	0.13%	32 kB	5,907

Table 5: Results of 10-fold cross-validation with evaluated parameters shows very good generalization properties (learning on 90%, and testing on remaining 10%)

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%

Dissemination

Current hyphenation support based on hyphenation patterns is collected in the `hyph-utf8` [17] project. The project uses ISO standards, notably Unicode and IETF language tags BCP 47. BCP 47 defines a `Scope` property to identify subtags for language collections. `hyph-utf8` currently contains hyphenation patterns for 65 different languages with an additional 9 dialect or transliteration variants.

Our new patterns for "the Czechoslovak language" were accepted for inclusion to the `hyph-utf8` repository [17], and will be supported in the next revisions of `hyph-utf8` and `polyglossia` in the TEX Live distribution. LuaTEX allows loading patterns at runtime, for only the languages used in a document. For other engines, *all* the patterns have to be loaded in precomputed, compact form into TEX's memory from the format file at the start of every document compilation.

As suggested by TEX experts, we prefer Czech and Slovak `\language`s being internally synonyms, with patterns only loaded once.

Using the patterns via available libraries in many programming languages (JavaScript, Perl, Python, C, and more) is straightforward and makes the patterns' usage versatile. Most typesetting systems and browsers, including OpenOffice and Chrome, could hyphenate in narrow columns of mobile devices. Most of them, if not all systems, use pattern technology and practices from the TEX community anyway.

We will support pattern dissemination in TEX distributions and multilingual support packages. We will tidy up available language resources with the community of Czech and Slovak users.

Future work

We think of developing language-agnostic patterns for syllabically hyphenated languages, based on available data from CELEX [13] with our workflow and evaluation measures. Wordpiece segmentation algorithm [30] gives superb results in the NLP domain for language translation, indicating that information is conveyed via character n-grams. With universal, syllable-based patterns, it will be possible to hyphenate text for most syllabically hyphenated languages even without knowing the language markup.

Another direction of research attention will be machine-learned heuristics for setting of `patgen` generation parameters, with the objective of metrics optimization used in the evaluation. When applied to the languages with available word lists, it would lead to pattern improvements for most supported languages.

Acknowledgment

We are indebted to Don Knuth for questioning the common properties of Czech and Slovak hyphenation during our presentation of [26] at TUG 2019, which has led us in this research direction. We also thank everyone on whose shoulders we build our work, and to all who commented on our workflow, patterns, and this paper, and discussed it at TUG 2021.

References

[1] R.E. Allen, ed. *The Oxford Spelling Dictionary*, vol. II of *The Oxford Library of English Usage*. Oxford University Press, 1990.

[2] S. Bartlett, G. Kondrak, C. Cherry. Automatic Syllabification with Structured SVMs for Letter-to-Phoneme Conversion. In *Proceedings of ACL-08: HLT*, pp. 568–576, Columbus, Ohio, June 2008. ACL. `https://www.aclweb.org/anthology/P08-1065`

[3] A. Bauer. *Dělení slov / slovotvorba v praxi /* [Word hyphenation / practical morphology /]. Nakladatelství Olomouc, Olomouc, 1997.

[4] *The Chicago Manual of Style*. University of Chicago Press, Chicago, 17th ed., Sept. 2017.

[5] J. Chlebíková. Ako rozdeliť (slovo) Československo [How to hyphenate the word Czechoslovakia]. *Zpravodaj $\mathcal{CS}$ TUG* 1(4):10–13, Apr. 1991. `10.5300/1991-4/10`

[6] P.B. Gove, M. Webster. *Webster's Third New International Dictionary of the English language Unabridged*. Merriam-Webster Inc., Springfield, Massachusetts, U.S.A, Jan. 2002.

[7] J. Haller. *Jak se dělí slova* [How Words Get Hyphenated]. Státní pedagogické nakladatelství Praha, 1956.

[8] Y. Haralambous. A Revisited Small Tutorial on Patgen, 28 Years After, Mar. 2021. `https://ctan.org/pkg/patgen2-tutorial`

[9] Internetová jazyková příručka [Internet Language Reference Book]. `https://prirucka.ujc.cas.cz/?id=135`

[10] M. Jakubíček, A. Kilgarriff, et al. The TenTen Corpus Family. In *Proc. of the 7th International Corpus Linguistics Conference (CL)*, pp. 125–127, Lancaster, UK, July 2013.

[11] M. Keary. On hyphenation—anarchy of pedantry. *PC Update, The magazine of the Melbourne PC User Group*, 2005. `https://web.archive.org/web/20050310054738/http://www.melbpc.org.au/pcupdate/9100/9112article4.htm`

[12] A. Kilgarriff, P. Rychlý, et al. The Sketch Engine. In *Proceedings of the Eleventh EURALEX International Congress*, pp. 105–116, Lorient, France, 2004.

[13] J. Krantz, M. Dulin, P.D. Palma. Language-Agnostic Syllabification with Neural Sequence Labeling. In *18th IEEE International Conference On Machine Learning And Applications, ICMLA 2019, Boca Raton, FL, USA, December 16–19, 2019*, M.A. Wani, T.M. Khoshgoftaar, et al., eds., pp. 804–810. IEEE, 2019. 10.1109/ICMLA.2019.00141

[14] F.M. Liang. *Word Hy-phen-a-tion by Com-put-er*. Ph.D. thesis, Department of Computer Science, Stanford University, Aug. 1983. https://tug.org/docs/liang/liang-thesis.pdf

[15] I. Maddieson. Syllable Structure. In *The World Atlas of Language Structures Online*, M.S. Dryer, M. Haspelmath, eds. Max Planck Institute for Evolutionary Anthropology, Leipzig, 2013. https://wals.info/chapter/12

[16] Y. Marchand, C.R. Adsett, R.I. Damper. Automatic Syllabification in English: A Comparison of Different Algorithms. *Language and Speech* 52(1):1–27, 2009. 10.1177/0023830908099881

[17] A. Rosendahl, M. Miklavec. TEX hyphenation patterns. http://hyphenation.org/tex

[18] Y. Shao, C. Hardmeier, J. Nivre. Universal Word Segmentation: Implementation and Interpretation. *Transactions of the Association for Computational Linguistics* 6:421–435, 2018. 10.1162/tacl_a_00033

[19] O. Sojka, P. Sojka. cshyphen repository. https://github.com/tensojka/cshyphen

[20] P. Sojka. Notes on Compound Word Hyphenation in TEX. *TUGboat* 16(3):290–297, 1995. https://tug.org/TUGboat/tb16-3/tb48soj2.pdf

[21] P. Sojka. Hyphenation on Demand. *TUGboat* 20(3):241–247, 1999. https://tug.org/TUGboat/tb20-3/tb64sojka.pdf

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

[23] P. Sojka. Slovenské vzory dělení: čas pro změnu? [Slovak Hyphenation Patterns: A Time for Change?]. *CS TUG Bulletin* 14(3–4):183–189, 2004. 10.5300/2004-3-4/183

[24] P. Sojka, O. Sojka. The Unreasonable Effectiveness of Pattern Generation. *Zpravodaj CS TUG* 29(1–4):73–86, 2019. 10.5300/2019-1-4/73

[25] P. Sojka, O. Sojka. Towards Universal Hyphenation Patterns. In *Proc. of Recent Advances in Slavonic Natural Language Processing—RASLAN 2019*, A. Horák, P. Rychlý, A. Rambousek, eds., pp. 63–68, Karlova Studánka, Czech Republic, 2019. Tribun EU. https://nlp.fi.muni.cz/raslan/2019/paper13-sojka.pdf

[26] P. Sojka, O. Sojka. The unreasonable effectiveness of pattern generation. *TUGboat* 40(2):187–193, 2019. https://tug.org/TUGboat/tb40-2/tb125sojka-patgen.pdf

[27] P. Sojka, O. Sojka. Towards New Czechoslovak Hyphenation Patterns. *Zpravodaj CS TUG* 30(3–4):118–126, 2020. https://cstug.cz/bulletin/pdf/2020-3-4.pdf#page=16

[28] P. Sojka, P. Ševeček. Hyphenation in TEX — Quo Vadis? *TUGboat* 16(3):280–289, 1995. https://tug.org/TUGboat/tb16-3/tb48soj1.pdf

[29] N. Trogkanis, C. Elkan. Conditional Random Fields for Word Hyphenation. In *Proc. of the 48th Annual Meeting of the ACL*, pp. 366–374, Uppsala, Sweden, July 2010. ACL. https://www.aclweb.org/anthology/P10-1038

[30] Y. Wu, M. Schuster, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, 2016. https://paperswithcode.com/method/wordpiece

[31] Ľ. Štúr Institute of Linguistics of the Slovak Academy of Sciences (SAS), ed. *Pravidlá slovenského pravopisu* [Rules of Slovak Grammar]. Veda, publisher of SAS, Bratislava, 3rd (updated printing) ed., 2000. https://www.juls.savba.sk/ediela/psp2000/psp.pdf

◇ Petr Sojka
Faculty of Informatics, Masaryk Univ., Brno, Czech Republic
sojka (at) fi dot muni dot cz
https://www.fi.muni.cz/usr/sojka/
ORCID 0000-0002-5768-4007

◇ Ondřej Sojka
Faculty of Informatics, Masaryk Univ., Brno, Czech Republic
454904 (at) mail dot muni dot cz
ORCID 0000-0003-2048-9977