Zpravodaj Československého sdružení uživatelů TeXu

Petr Sojka; Ondřej Sojka

Towards New Czechoslovak Hyphenation Patterns

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 30 (2020), No. 3-4, 118–126

Persistent URL: http://dml.cz/dmlcz/150285

Towards New Czechoslovak Hyphenation Patterns

Petr Sojka, Ondřej Sojka

Space- and time-effective segmentation and hyphenation of natural languages stay at the core of every document preparation system, web browser, or mobile rendering system. Recently, the unreasonable effectiveness of pattern generation has been shown – it is possible to use hyphenation patterns to solve the dictionary problem for a single language without compromise. In this article, we will show how we applied the marvelous effectiveness of `patgen` for the generation of the new Czechoslovak hyphenation patterns that cover two languages. We show that the development of more universal hyphenation patterns is feasible, allows for significant quality improvements and space savings. We evaluate the new approach and the new Czechoslovak hyphenation patterns.

Keywords: hyphenation, hyphenation patterns, `patgen`, syllabification, syllabic hyphenation, Czech, Slovak, Czechoslovak patterns

"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 [1]

Introduction

Hyphenation is at the core of every document preparation system, be it TEX or any modern web browser. It has been shown [2] that data-driven approaches to hyphenation and syllabification algorithms outperform rule-based ones, reaching accuracy around 95% per a single language. Bartlett et al. [3] developed a machine learning approach for automatic syllabification, motivated by the needs of letter-to-phoneme conversion. Trogkanis et al. [4] used conditional random fields for word hyphenation, and compared the accuracy and other metrics with the original technique of Liang [5]. Their results abstracted from heuristics to optimize generated patterns by `patgen`, diminishing achievable performance.

There are about 5,000 languages supported by Unicode Consortium that are still in use today. In a digital typographic system that supports Unicode and its languages in full, there should be support for algorithms, rules, or language hyphenation patterns. Recently, there were attempts to tackle the word segmentation problem in different languages by Shao et al. [6]. The algorithm is error-prone, but it was developed primarily for speech recognition and language representation tasks, where a small number of errors is tolerated. On the contrary, in a typesetting system like TEX, errors in hyphenation are not tolerated at all—all exceptions have to be covered by the algorithm.

 doi: 10.5300/2020-3-4/118

Current typesetting support in the TEX Live distribution contains [7] hyphenation patterns for about 80 different languages. All these patterns have to be loaded into TEX's memory at the start of every compilation, which slows down compilation.

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 border of the stem and an affix, prefix or negation. A typical example are the British hyphenation rules by the Oxford University Press [8].

phonology-based Hyphenation based on the pronunciation of syllables allows reading text with hyphenated lines almost as if the hyphenation were not there. This pragmatic approach is preferred by the American publishers [9] and the Chicago Manual of Style [10].

In this paper, we evaluate 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 [11, 12, 13] and Slovak [14]. We document our reproducible workflow and resources in a public repository.

"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 [1]

Methods

The core idea is to develop common hyphenation patterns for phonology-based languages. In the case that these languages share pronunciation rules, homographs from different languages typically do not cause problems, as they are hyphenated the same. The very rare cases, where hyphenation is dictated by the seam of compound word contrary to phonology (`roz-um` vs. `ro-zum`), could be simply solved by not allowing the hyphenation of this particular word around this particular seam.

Recently, it was shown that the approach to generate hyphenation patterns from word list by program `patgen` is unreasonably effective [15]. One can set the parameters of the generation process so that the patterns cover 100% of hyphenation points, and the size of the patterns remains reasonably small. For the Czech language, hyphenation points from 3,000,000 hyphenated words are squeezed into 30,000 bytes of patterns, as stored in the compressed trie data structure. That means achieving a compression ratio of several orders of magnitude with 100% coverage and nearly zero errors [15]. For a similar language such as Slovak, the pronunciation is very similar, syllable-forming principles are the same, and also 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. That means that *only* words like `oblít` will not be hyphenated, because the typesetting system currently cannot decide in which meaning the word is used: `o-blít` or `ob-lít`.

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

Data Preparation

For our work, word lists with frequencies for Czech and Slovak were donated by Lexical Computing CZ from the TenTen family of corpora [16, 17].

The Czech word list was cleaned up and extended as described by Sojka et Sojka [15, 18], using the Czech morphological analyzer `majka`. Only words that occurred more than ten times were used in further processing. The final word list `cs-all-cstenten.wls` contained 606,494 words.

For Slovak, we obtained 1,048,860 Slovak words with frequency higher than ten from 2011 SkTenTen corpora [16]. We only used words with a frequency higher than thirty that comprised only of ISO Latin 2 characters, obtaining 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`.

Pattern Development

The workflow of the Czechoslovak pattern development is illustrated in Figure 1 on the facing page. We have used recent accurate Czech patterns [15] for the hyphenation of the joint Czech and Slovak word list. We had to manually fix incorrect hyphenation, typically near the prefix and stem of words when phoneme-based hyphenation point was one character away from the seam of the prefix or compound word: `neja-traktivnější`, `neja-teističtější`, `neje-kologičtější`.

We have then hyphenated words used in both languages also by 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 an input to `patgen` with a higher weight during generation of the final Czechoslovak hyphenated patterns.

It must be noted that 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 of Sojka et Sojka [21]. Gentle reader may also find the scripts used there.

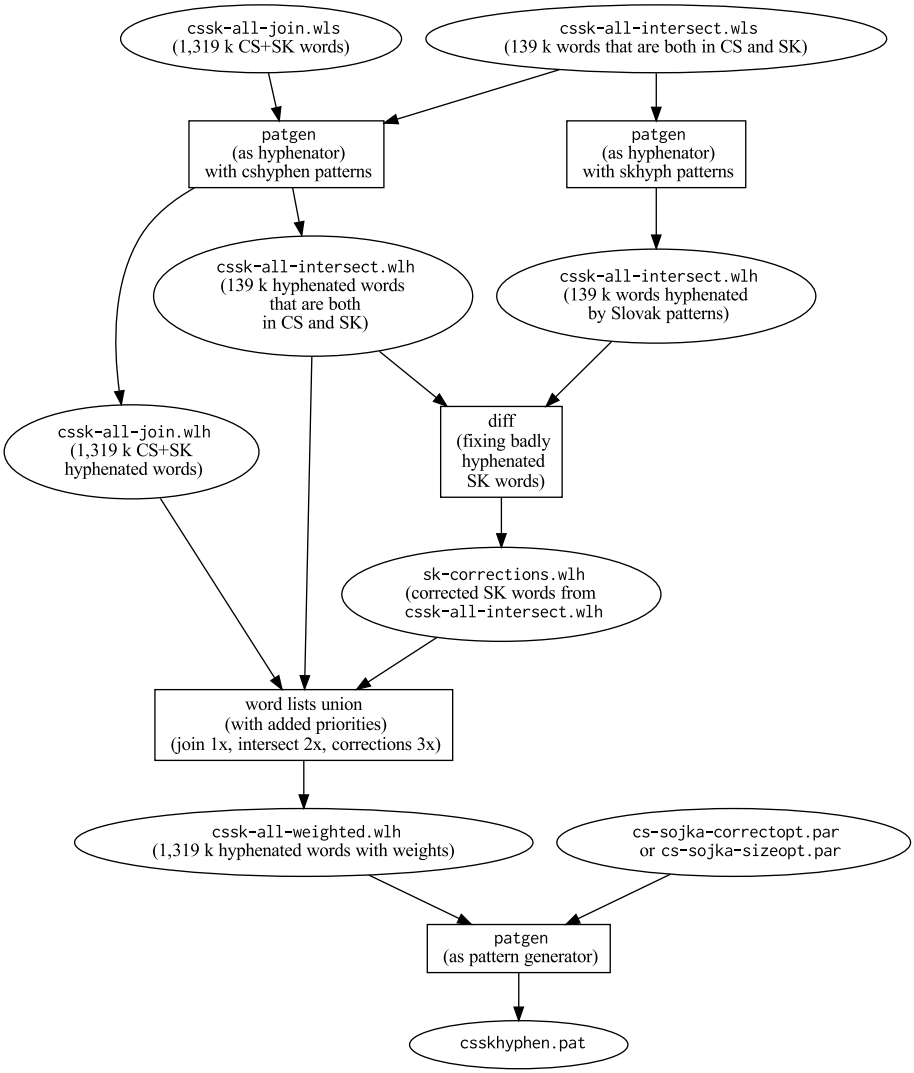

Figure 1: The development process of the new Czechoslovak patterns: Bootstrapping with Czech patterns, checking and fixing with a higher weight Slovak words that are common with Czech ones.

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: 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	[19, by hand]	N/A	N/A	N/A	20 kB	2,467
Czech	correctopt [15]	99.76%	2.94%	0.24%	30 kB	5,593
Czech	sizeopt [15]	98.95%	2.80%	1.05%	19 kB	3,816
Slovak	[20, Table 1, `patgen`]	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 4: Results of 10-fold cross-validation with evaluated parameters

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%

Evaluation

The evaluation of the quality of developed patterns could be done as an evaluation of *coverage* of hyphenation points in the training wordlist—how many hyphenation points used in training were correctly predicted by the patterns—and of *generalization* properties—how the patterns behave on unseen data, on the words not available in the data used during `patgen` training.

Both coverage and generalization could be viewed as a *classification* task, i.e. how the patterns classify hyphenation points in the training and testing wordlists, respectively.

Classification

For evaluation of classification, 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– 3 on the preceding page, we report: **Good** sum or percentage of found hyphenation points as a sum of TP and TN, **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 clearly more severe than type 2 errors in our hyphenation points setup. Nonzero **Bad** results do not necessarily mean that the patterns performed badly, the opposite is often the case—patterns have found a rule that is not obeyed in the ground truth wordlist. In other words, they found an inconsistency that needs to be fixed in the underlying wordlist, rather than a valid exception. This practice of manually inspecting bad hyphenation points has been used during the development of the wordlist.

Generalization

To assess the generalization properties, we used 10-fold cross-validation, leaving one tenth out of the training set to evaluate the effectiveness of the patterns on unseen words. The results are shown in the Table 4 on the facing page. The evaluation metrics slightly differ with different `patgen` parameters, with best results achieved when coverage of the training set is maximized.

The achieved results show that both evaluation metrics are close to perfection, as we are free to either push for perfect coverage, and reach it (lossless compression of wordlist hyphenation points by the developed pattern), or push to maximize generalization qualities, and miss only less than 1% of valid hyphenation points. Doing that for two languages in parallel seems like a good result.

"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 [1]

Conclusion and Future Works

We have shown that the development of common hyphenation patterns for languages with similar pronunciation is feasible. `Patgen` was able to generalize hyphenation rules common for both languages with 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, see a *Jupyter demo notebook* and all **source code** by Sojka et Sojka [21].

We will offer the new patterns for "the Czechoslovak language" to the TEX Live distribution, creating the first language support package to be shared by multiple languages. With this route, the new patterns will appear in most typesetting systems and browsers including OpenOffice and Chrome quickly, as they use the pattern technology and patterns from the TEX community (the `tex-hyphen` repository).

Acknowledgement
We are indebted to Don Knuth for questioning the common properties of Czech and Slovak hyphenation during our presentation of [15] at TUG 2019, which has led us in this research direction.

References

1. KEARY, Major. On Hyphenation – Anarchy of Pedantry. *PC Update, The magazine of the Melbourne PC User Group*. 2005. Available also from: `https://web.archive.org/web/20050310054738/http://www.melbpc.org.au/pcupdate/9100/9112article4.htm`.
2. MARCHAND, Yannick, ADSETT, Connie R. et DAMPER, Robert I. Automatic Syllabification in English: A Comparison of Different Algorithms. *Language and Speech*. 2009, vol. 52, no. 1, pp. 1–27. Available from DOI: `10.1177/0023830908099881`.
3. BARTLETT, Susan, KONDRAK, Grzegorz et CHERRY, Colin. Automatic Syllabification with Structured SVMs for Letter-to-Phoneme Conversion.

In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, 2008, pp. 568–576. Available also from: https://www.aclweb.org/anthology/P08-1065.

4. TROGKANIS, Nikolaos et ELKAN, Charles. Conditional Random Fields for Word Hyphenation. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 366–374. Available also from: https://www.aclweb.org/anthology/P10-1038.
5. LIANG, Franklin M. *Word Hy-phen-a-tion by Com-put-er*. 1983. PhD thesis. Stanford University.
6. SHAO, Yan, HARDMEIER, Christian et NIVRE, Joakim. Universal Word Segmentation: Implementation and Interpretation. *Transactions of the Association for Computational Linguistics*. 2018, vol. 6, pp. 421–435. Available from DOI: 10.1162/tacl_a_00033.
7. REUTENAUER, Arthur et MIKLAVEC, Mojca. *TEX hyphenation patterns*. TUG, [n.d.]. Available also from: https://tug.org/tex-hyphen/. Accessed 2019-11-24.
8. ALLEN, R. E. (ed.). *The Oxford Spelling Dictionary*. Oxford University Press, 1990. The Oxford Library of English Usage.
9. GOVE, Philip Babcock et WEBSTER, Merriam. *Webster's Third New International Dictionary of the English language Unabridged*. Springfield, Massachusetts, U.S.A: Merriam-Webster Inc., 2002.
10. ANONYMOUS. *The Chicago Manual of Style*. 17th ed. Chicago: University of Chicago Press, 2017. ISBN 9780226287058.
11. SOJKA, Petr. Notes on Compound Word Hyphenation in TEX. *TUGboat*. 1995, vol. 16, no. 3, 290–297. Available also from: https://tug.org/TUGboat/tb16-3/tb48soj2.pdf.
12. SOJKA, Petr et ŠEVEČEK, Pavel. Hyphenation in TEX—Quo Vadis? *TUGboat*. 1995, vol. 16, no. 3, 280–289. Available also from: https://tug.org/TUGboat/tb16-3/tb48soj1.pdf.
13. SOJKA, Petr. Hyphenation on Demand. *TUGboat*. 1999, vol. 20, no. 3, 241–247. Available also from: https://tug.org/TUGboat/tb20-3/tb64sojka.pdf.
14. SOJKA, Petr. Slovenské vzory dělení: čas pro změnu? (Slovak Hyphenation Patterns: A Time for Change?) *CSTUG Bulletin*. 2004, vol. 14, no. 3–4, 183–189. Available from DOI: 10.5300/2004-3-4/183.
15. SOJKA, Petr et SOJKA, Ondřej. The unreasonable effectiveness of pattern generation. *TUGboat*. 2019, vol. 40, no. 2, pp. 187–193. Available also from: https://tug.org/TUGboat/tb40-2/tb125sojka-patgen.pdf.
16. JAKUBÍČEK, Miloš, KILGARRIFF, Adam, KOVÁŘ, Vojtěch, RYCHLÝ, Pavel et SUCHOMEL, Vít. The TenTen Corpus Family. In: *Proc. of the*

7th International Corpus Linguistics Conference (CL). Lancaster, 2013, pp. 125–127.
17. KILGARRIFF, Adam, RYCHLÝ, Pavel, SMRŽ, Pavel et TUGWELL, David. The Sketch Engine. In: *Proceedings of the Eleventh EURALEX International Congress*. Lorient, France, 2004, pp. 105–116.
18. SOJKA, Petr et SOJKA, Ondřej. The Unreasonable Effectiveness of Pattern Generation. *Zpravodaj ČSTUG*. 2019, vol. 29, no. 1–4, 73–86. Available from DOI: 10.5300/2019-1-4/73.
19. CHLEBÍKOVÁ, Jana. Ako rozdeliť (slovo) Československo (How to hyphenate the word Czechoslovakia). *Zpravodaj ČSTUG*. 1991, vol. 1, no. 4, 10–13. Available from DOI: 10.5300/1991-4/10.
20. SOJKA, Petr. Slovenské vzory dělení: čas pro změnu? In: *Proceedings of SLT 2004, 4th seminar on Linux and TEX*. Znojmo: Konvoj, 2004, 67–72. Available also from: https://fi.muni.cz/usr/sojka/papers/skhyp.pdf.
21. SOJKA, Ondřej et SOJKA, Petr. *cshyphen repository*. [N.d.]. Available also from: https://github.com/tensojka/cshyphen.

Na cestě k novým československým vzorům dělení

Prostorově a časově efektivní segmentace a dělení slov přirozených jazyků zůstává jádrem každého systému pro přípravu dokumentů, webového prohlížeče nebo zlomu dokumentů na mobilních zařízeních. Nedávno jsme ukázali obrovskou účinnost generování vzorů a bylo prokázáno, že je možné použít vzory dělení slov k vyřešení slovníkového problému (automatické segmentace) pro jeden jazyk bez kompromisů (100% pokrytí). V tomto článku ukazujeme, jak jsme použili úžasnou účinnost patgenu pro generování vzorů dělení slov, které pokrývají dva jazyky zároveň, pro nové, společné vzory československého dělení. Ukazujeme, že je možné vyvinout univerzálnější vzory dělení slov, což umožňuje jak kvalitativní zlepšení, tak i úsporu místa oproti předchozí dvojici vzorů pro jednotlivé jazyky. Hodnotíme nový přístup a nové společné československé vzory dělení.

Klíčová slova: patgen, vzory dělení slov, československé dělení, efektivní segmentace, slabičné dělení pro více jazyků

Petr Sojka, ORCID: 0000-0002-5768-4007
Faculty of Informatics, Masaryk University
Brno, Czech Republic and ČSTUG, sojka@fi.muni.cz

Ondřej Sojka, ORCID: 0000-0003-2048-9977
Faculty of Informatics, Masaryk University
Brno, Czech Republic, 454904@mail.muni.cz