Expanding Hyphenation Patterns Across Slavic Languages

Ondřej Sojka

Abstract

So far, TeX hyphenation patterns, even for related languages, have been developed separately for each language, splitting scarce human resources. As languages develop and especially English terms creep into formerly monolingual texts, hyphenation patterns, especially for medium- and low-resource languages which often lack quality generated patterns, are due for an update. In this article, we explore the possibilities for transfer learning of hyphenation rules between related Slavic languages.

We present new hyphenation patterns for multiple Slavic languages, developed using transfer learning from various sources.

1 Motivation

Hyphenation patterns play a crucial role in typesetting and text layout, particularly for languages with long words or narrow text columns. They ensure proper word breaks at line ends, improving readability and aesthetics of printed or digital text. Good hyphenation patterns contribute to more uniform text distribution, reducing the occurrence of large gaps between words or excessive hyphenation, making reading more pleasant.

The quality of hyphenation patterns across Slavic languages varies, with low- and medium-resource languages being impacted the most. Often, the only patterns available are ones made by hand, without the pattern generation program `Patgen` [2], more than a decade ago. These are insufficient, especially considering the mediocre generalization capabilities of `Patgen`.

Hyphenation patterns in Slavic languages are, however, *syllabic* and syllabification is very similar across languages.

Pattern generation is also a niche topic and the associated know-how is fairly sparsely distributed.

But since syllabification rules and patterns do not vary across languages from the same family, why do we have to develop patterns for each language separately? After all, native speakers of one Slavic language, upon *hearing a spoken word* from a different Slavic language and being provided with the written form, can hyphenate it correctly.

If we can acquire text data that express the spoken form of the word, we should be able to generate patterns that hyphenate as well as such a native speaker.

2 International Phonetic Alphabet

The International Phonetic Alphabet (IPA) [1] is a standardized system for representing the sounds of human speech. Created by the International Phonetic Association, it uses Latin-based symbols to uniquely represent phonemes, stress, and intonation across all languages. In our project, IPA serves as a crucial intermediary, providing a **common phonetic representation** that *bridges orthographic differences* between Slavic languages. This allows us to capture phonological similarities that might be obscured by orthographic differences and varied scripts (Latin vs. Cyrillic), enabling effective cross-linguistic transfer of hyphenation patterns.

3 Joint IPA-form data preparation

3.1 Data acquisition

To start, we need a dataset of words used in each of the *target languages*[1] with frequency data. Given the importance of replicability and licensing restrictions often placed on proprietary datasets, we settled with a cleaned wordlist of all words from the Wikipedia of each language. We strip the XML tags and clean words that occur relatively more frequently on Wikipedia as part of common article layouts, such as Table, References, External links and similar, acquiring a *replicable*, relatively clean, wordlist.

3.2 Hyphenation of the original word forms

We apply the best available hyphenation patterns for each target language to hyphenate all the words in our frequency word list with a frequency higher than 100 and generate the file `{lang}.wlh`.

3.3 Transferring hyphens to the IPA word form

We use `espeak-ng` [4] to convert from the written word form (in either Latin or Cyrillic script) to the form in International Phonetic Alphabet (IPA) [1].

The next step is to acquire hyphenated words in IPA and *transfer* the hyphens from the written (Latin or Cyrillic) form to IPA. We use Algorithm 1 on the following page to transfer the hyphens.

This approach is very computationally expensive, but it is very parallelizable and therefore not a problem on modern hardware.

[1] Target languages are all Slavic languages for which some hyphenation patterns currently exist and which have their own mutation of Wikipedia. Only languages which pass evaluation will be proposed for inclusion in `hyph-utf8` [3]

Algorithm 1 Transfer Hyphens Between Word Forms

Require: hyphenated (hyphenated word in source script), target (unhyphenated word in target script)
Ensure: best_result (hyphenated word in target script)

```
function TransferHyphens(hyphenated, target)
    num_hyphens ← CountHyphens(hyphenated)
    possible_positions ← {1,..., len(target) −1}
    best_result ← ""
    min_distance ← ∞
    for hyphen_positions in Combinations(possible_positions, num_hyphens) do
        if First(hyphen_positions) ≠ 0 and Last(hyphen_positions) ≠ len(target) −1 then
            candidate ← InsertHyphens(target, hyphen_positions)
            current_distance ← LevenshteinDistance(hyphenated, candidate)
            if current_distance < min_distance then
                best_result ← candidate
                min_distance ← current_distance
            end if
        end if
    end for
    return best_result
end function
```

4 Joint IPA-form pattern generation

To generate patterns that hyphenate across languages in IPA, we need to first decide what data to use. If we were to weigh data from each language in the training set equally, considering that any machine learning model generally can be only as good as its training data (in the absence of advanced techniques), we would get mediocre patterns.

We weigh the languages based on the quality of hyphenation in their data, ensuring inclusion of every language while considering the overlap of IPA characters used across languages. We will present more details on the training data mix in the final version of this paper.

5 Final language-specific pattern generation

As the final step, we convert each of the target language frequency datasets to IPA, hyphenate them with the joint patterns and use the algorithm 1 to transfer the hyphens to the target language. Having a well-hyphenated wordlist, we run `Patgen` with the `cs-sojka-correctoptimized.pat` [5] patterns and generate the final language-specific patterns.

6 Evaluation

To evaluate the quality of the resulting patterns, we turn from machines back to humans. Native speakers of every target language will be presented with sets of 100 randomly shuffled hyphenations and will be asked to decide which hyphenation they find better. For languages in which the improvement has cleared the threshold of statistical significance, we will propose their inclusion into `tex-hyphen` [3], the de facto canonical repository of hyphenation patterns.

References

[1] I.P. Association. *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, Cambridge, 1999.

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

[3] A. Reutenauer, M. Miklavec. TEX hyphenation patterns. Accessed 2024-06-25. `https://hyphenation.org/`

[4] J. Reynolds. eSpeak NG, 2016. `https://github.com/espeak-ng/espeak-ng`

[5] P. Sojka, O. Sojka. New Czechoslovak Hyphenation Patterns, Word Lists, and Workflow. *TUGboat* 42(2), 2021. `https://doi.org/10.47397/tb/42-2/tb131sojka-czech`

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