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 available 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.

doi.org/10.47397/tb/45-2/tb140sojka-slavic

Algorithm 1 Transfer hyphenations 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 $\leftarrow$ CountHyphens(hyphenated)
    possible_positions $\leftarrow \{1,\ldots,$ len(target) $-1\}$
    best_result $\leftarrow$ ""
    min_distance $\leftarrow \infty$
    for hyphen_positions in Combinations(possible_positions, num_hyphens) do
        if First(hyphen_positions) $\neq 0$ and Last(hyphen_positions) $\neq$ len(target) $-1$ then
            candidate $\leftarrow$ InsertHyphens(target, hyphen_positions)
            current_distance $\leftarrow$ LevenshteinDistance(hyphenated, candidate)
            if current_distance $<$ min_distance then
                best_result $\leftarrow$ candidate
                min_distance $\leftarrow$ current_distance
            end if
        end if
    end for
    return best_result
end function
```

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-lingual 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 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 50 and generate the file ⟨*lang*⟩`.wlh`, containing the **w**ord **l**ist **h**yphenated.

3.3 Transfer of hyphenations to 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 IPA [1].

The next step is to acquire hyphenated words in IPA by *transferring* the hyphenations from the written (Latin or Cyrillic) form to IPA. We use Algorithm 1 to transfer the hyphenations.

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

4 Joint IPA-form pattern generation

To generate patterns that hyphenate across languages in IPA, we first need to 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, we would get mediocre patterns.

[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].

Patterns are indeed able to learn the IPA dataset, as shown by the results of a run with correct optimized parameters: good: 99.84%, bad: 0.13%, missed: 0.16%.

4.1 Ground truth data for evaluation of data mixes

To decide on the mix, we need data to evaluate the quality of a given language-specific pattern set. To do this, we acquire ground truth data from various sources — in order of preference: language institutes, dictionaries, wiktionaries, human labelers, etc. It is disappointingly rare to find hyphenations in orthographic dictionaries.

4.2 Mixing training data for joint IPA-form pattern generation

To generate high-quality patterns that effectively hyphenate across Slavic languages in IPA form, we employ a strategic approach to mixing training data. Our process involves the following steps:

1. **Initial sampling:** We randomly sample five sets of weights from the possible weight set. Each weight corresponds to the importance given to a specific language's training data.
2. **Model fitting:** Using these initial weight sets, we fit a random forest model. This model learns the relationship between the weight combinations and the quality of the resulting patterns.
3. **Guided sampling:** The random forest model is then used to guide further sampling of weight combinations. This approach allows us to explore the weight space more efficiently, focusing on areas that are likely to yield better results.
4. **Evaluation:** For each set of weights, we generate patterns and evaluate them using a custom scoring function. The score is calculated as good − bad × 5, where 'good' represents correctly placed hyphenation points and 'bad' represents incorrectly placed ones. This scoring method heavily penalizes incorrect hyphenations while rewarding correct ones.
5. **Selection:** After exploring a predetermined number of weight combinations, we select the set that produces the highest score.

This method allows us to efficiently search the space of possible weight combinations and find an optimal mix of training data from different Slavic languages.

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 algorithm 1 on the previous page to transfer the hyphens to the target language. Having a well-hyphenated wordlist, we run Patgen with a custom parameter set inspired by previously published correct-optimized patterns [5]. 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] International Phonetic 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. `tug.org/docs/liang/liang-thesis.pdf`

[3] A. Reutenauer, M. Miklavec. TEX hyphenation patterns. `hyphenation.org/`

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

[5] P. Sojka, O. Sojka. New Czechoslovak hyphenation patterns, word lists, and workflow. *TUGboat* 42(2):152–158, 2021. `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