# emPhon: Morphologically sensitive open-source phonetic transcriber

Kulcsár Virág[1]*, Lévai Dániel[12]*

[1]HAS-BME Lendület Language Acquisition Research Group
[2]Alfréd Rényi Institute of Mathematics
{kulvirag@gmail.com, levaid@renyi.hu}

**Abstract.** We propose a new emtsv module which can provide phonetic transcription based on the emMorph state-of-the-art morphological analyzer. In the first part of the paper, we present the motivation, the main problem and the method we are using to leverage Hungarian phonetic transcription with the use of morphological analyses. The second part is about evaluation both intrinsically and extrinsically – we evaluate our transcriber based on the IPA transcriptions of Wiktionary and as part of a speech synthesizer system. The code and models are fully open-source and are available under LGPL 3.0 license at `https://github.com/levaid/emPhon`.

**Keywords:** phonetics, IPA, emtsv, morphology, analysis, speech, synthesis

## 1 Introduction, motivation

Hungarian is a highly phonetic language in its orthography. Compared to English, where the lack of productive morphology and the highly irregular orthography makes it hard for any rule-based transcriber to work efficiently, Hungarian orthography is much more informative. For a great extent, simple words like ‘alma‘ *apple* are easy to phonetically transcribe: we can look up the IPA for Hungarian and just substitute the letters one-by-one, giving /ɒlmɒ/ as pronunciation. In most cases, however, it is needed to take into account the Hungarian phonology - there are large set of rules which are governed by not only the interaction of different phones, but by the morphology as well.

The simplest problem that a phonetic transcriber has to overcome in Hungarian is the handling of digraphs (e.g. *cs*, *ny*) and their long counterparts. This is not a trivial task in itself, even though it looks like an innocent problem. One could argue that a modern subword tokenizer is easily able to distinguish between digraphs and and co-occurrences, but that is simply not true. Consider the following example, where the *sz* has been segmented into -*s*_ *z*-: *als_ zo_ tok* ‘sleep-2PL’, produced both by HuBERT’s tokenizer (Nemeskey, 2020) and by XLM-Roberta’s (Conneau et al., 2020) tokenizer. This example demonstrates

* equal contribution

well that rule-based methods can indeed be better in a (decreasingly small) number of NLP tasks than neural or unsupervised methods.

The more complex problems arise when the morphology creates letter clusters which are pronounced differently based on the underlying morphology. Let us take the sentence *Tűnj el, tűnj el te vagy az igazi bűnjel*[1] and inspect the -*nj*- segments in *tűn-j* 'disappear-2SG.IMP' and *bűn-jel* 'crime-sign.NOM'. The -*nj* segment at the end of *tűn-j* is pronounced as a voiced palatal nasal /ɲ/ due to the palatal assimilation in Hungarian. However, this rule does not hold in case of an inner (or an outer) word boundary: in *bűn-jel*, the -*nj*- cannot assimilate into /ɲ/, thus pronounced as /nj/ – the phones must be pronounced separately due to the previously mentioned inner word boundary.

A grapheme-to-phoneme transcriber which is able to solve the previously mentioned problems would be quite beneficial in speech systems or in phonetic research for example, and there is a lack of publicly available, open-source tool of this kind.

Given the above reasons and the versatility of the emtsv framework, we propose a new emtsv (Indig et al., 2019) module which would be of great use to the Hungarian NLP community.

## 2 Related work

Hungarian phonology, and more specifically, the interaction of pronunciation and orthography is a well-research subject. There are earlier, rule-based systems, even for Hungarian (Novák and Siklósi, 2016)[2], and there are machine learning-based methods, nowadays mostly neural-based techniques (Yolchuyeva et al., 2019), we have chosen to implement a rule-based system based on the theory of Hungarian phonology presented in Siptár and Törkenczy (2000).

Implementing those rules is not straightforward, since there is a lot of variation based on speed, style, dialect, and on the speaker's unique speech.

There is also a problem with the International Phonetic Alphabet for Hungarian: it is very inconsistent, as further explained in section 2.1.

Due to these obstacles, we have decided for a middle-ground approach, where the output produced by our transcriber represent the pronunciation generally well but may fail to cover edge-cases. In the final version of the emPhon, however, we are including these rules as user-configurable, as to customize and to adapt to the domain in which the tool is used.

### 2.1 IPA for Hungarian

There are multiple versions of Hungarian IPA in circulation with various levels of accuracy. In this subsection, we go through the most distinctive differences of the transcriptions.

[1] `https://zeneszoveg.hu/dalszoveg/31311/kontroll-csoport/keresnek-zeneszoveg.html`

[2] Sadly, neither the model nor the data is available from this paper, so we are unable to compare our performance to theirs.

In the vowel system, there is one difference between various papers and IPA that is worth noting. Some consider the grapheme *a* to be a low back rounded vowel /ɒ/ (e.g. Szende (1994)), while others argue that it is a low-mid back rounded vowel /ɔ/ (e.g. Siptár and Törkenczy (2000); Gósy (2004)). Considering that we measure on Wiktionary data, we chose it to be low back, but it can easily be substituted in the final output with any text processing tool.

The system of consonants is more complicated. There is a multitude of problems which could be addressed here and above all of that, a lot of rules are dependent on the style, speed, and speaker. We will focus on the most frequent and practical difficulties that we have encountered.

Concerning the stop-fricative clusters, we do not distinguish these from affricates, giving us *cím* /t͡siːm/ àddress, title' and *rendszer* /rent͡ser 'system', with *dsz* and *c* having identical representations. Even Wiktionary does not agree with itself; one[3] Wiktionary page distinguishes between these, while another[4] page does not.

Another problem in Hungarian IPA, the lack of marker for nasalization, even though it clearly happens, as in *tonhal* 'tuna' is represented as /tonhɑl/, instead of /tõːhɑl/.

The ambiguity of Hungarian IPA will further be addressed in section 4.

## 3 Transcriber

Our code can be divided into two distinct processes. The first takes a morphologically analyzed text and creates segmented text with special delimiters. The second is where the actual transcribing is happening. In this step we create an inner representation for ease of regex matching and as the last step, we convert it into Hungarian IPA.

### 3.1 Morphological segmentation

Our input is morphologically analyzed text. For this we used the morphological analyzer of the emtsv framework (emMorph, Novák et al. (2016)). The output of emMorph is every possible analysis for the given word. Unfortunately there is no disambiguation between lemma analyses, so our workaround was to use the *finest* splitting. We defined the *finest* splitting as the one containing the most morphs (for example *ár_ ad_ ás* is finer than *árad_ ás*, even when the former is highly unlikely) and that has proven to be acceptable baseline.

Segmentation was needed to avoid the pitfalls presented by words such as *kilenc-száz* 'nine-hundred' or *pác-só* 'marinade-salt'. Here the letter 'csz' could be divided as *c-sz* or *cs-z* and while native speakers can easily decide which one is correct, the computer is lost without additional help.

[3] `https://en.wiktionary.org/wiki/Appendix:Hungarian_pronunciation`

[4] `https://en.wikipedia.org/wiki/Help:IPA/Hungarian`

That is why we enlisted four different separators. The characters used are: |, #, §, ~, and they all signify different types of transitions that can be seen in table 1.

| Sep. | Transition | Example |
|---|---|---|
| # | between roots in compound words | rend#szer |
| ~ | between root and first suffix | emészt~és |
| § | between prefix and root | ki§mond |
| \| | between affixes | anyag~ok\|at |

**Table 1.** Separators and their transitions

By default, having four different separators is superfluous, but there might be edge-cases where the extra information provided by these separators can prove to be useful.

### 3.2 Transcription rules

After morphologically segmenting the text we can start transcribing. We based our pronunciation rules on Siptár and Törkenczy (2000), and implemented most of them using regex. This keeps the code pure python and provides a smoother installation experience while keeping the coding rather straightforward and running time low.

To handle digraphs and long phonemes, we created a unicode inner representation where every character defines exactly one sound. We decided to use the uppercase letters to stand for geminates in our inner representation. The double letter function has to be the first one to run in order to properly preprocess text. The long letter function runs as the finishing step of every other function so if another rule creates double consonants, they are handled. Below is the overview of different rules. To see the exact rules, the similarly named functions in the code should be referenced. The tool is implemented as a Python class with methods as rules, thus the order and the need for multiple runs can be easily achieved.

- **Digraphs**: The digraphs are substituted at the beginning of the process.
- **Degemination**: There are 4 different types of geminates based on the order of the geminate and the consonant and on whether the geminate is segmented or not.
- **Hiatus filling**: Hiatuses are filled in with /j/ in Hungarian.
- **Nasal place assimilation**: The nasal consonants assimilate based on the location of the consonant after it.
- **Elision of n**: The /n/ fully assimilates with the non-nasal sonorants after.
- **Elision of l**: The /l/ fully assimilates if there is /r/ after.
- **Palatal assimilation**: There are some consonants which are sensitive to palatalization and if these are adjacent to a palatal, they interact.

- **Sibilant rules**: There are two larger sets of rules here. One treats the sibilant fricative clusters, the other treats stop + sibilant clusters.
- **/h/-alternations**: The alternation of /h/ is a very complex phenomenon. There is a high degree of variance from speaker-to-speaker, thus we aimed to maximize the coverage on our evaluation dataset using a very limited number of rules.
- **Voice assimilation**: In Hungarian, we have regressive voicing assimilation, meaning that the voice of an obstruent cluster is governed by the rightmost consonant.

The last step is to convert the inner representation into standardized IPA.

# 4 Evaluation

We evaluate our phonetic transcriber two ways: first, we compare it with gold-standard phonetic transcriptions downloaded from the Hungarian Wiktionary[5]. Second, we use it to leverage a text-to-speech system on a single-speaker dataset.

## 4.1 Wiktionary

Wiktionary is a sister-project of Wikipedia, it is a multilingual, free content dictionary, it is run by the Wikimedia Foundation and is written by volunteers. Extracting the phonetic data was not straightforward: apparently the phonetic transcriptions are not saved in the periodic Wiktionary dumps[6], thus we had downloaded 65586 pages one-by-one and extracted the pronunciations using regular expressions.

The gold-standardness of this Wiktionary data is highly arguable based on our observations. There are systematic errors which indicate that there is some kind of automatic transcriber writing these. Below, we give an overview of the common errors.

One of the common errors is the creation of digraphs through morphological segments.

- vízsugár: /viːʒugaːr/
- gazság: /gɒʒaːg/

Using hiatus filling unnecessarily is also a common error.

- látóideg: /laːtoːjidɛg/
- adójóváírás: /ɒdoːjoːvaːjiːraːʃ/

The pronunciation of numbers is also questionable in places.

- harminchét: /hɒrmint͡sɦet/

[5] https://hu.wiktionary.org/

[6] https://dumps.wikimedia.org/huwiktionary/

- ezerkilencszázharminchat: /ezerkilend͡ʒzaːshɒrminɦɒt/

There is a general problem with the inclusion of foreign proper nouns.

- Auschwitz: /ɒuʃxvidz/
- Schmahl: /ʃxmaxl/

We have also encountered errors caused by some words redirecting to other words' page, and therefore having the wrong pronunciation associated with it. This mostly happened with slang words of crude meaning. In order to avoid this we decided to dismiss the words which had a pronunciation considerably longer or shorter than the original word length. In the end we were left with 64666 data points.

| Morph | h-deletion | WER |
|---|---|---|
| Yes | No | 4.10 % |
| Yes | Yes | 3.25 % |
| No | No | 3.71 % |
| No | Yes | 2.86 % |

**Table 2.** Word Error Rates

It will be quite hard to achieve lower WER on Wiktionary due to the problems above. As there are systematic errors in the last few percents, one has to decide the target of optimization: the Wiktionary data or the actual pronunciation. Some of the remaining problems can be rectified by creating a pronunciation lexicon which details how words should be written to reflect Hungarian pronunciation.

We also measured the WER by eliminating the postlexical /h/ variants and rules both in the dataset and in the transcriber. This is justified by the complexity and marginality of the Hungarian /h/.

We also measured the effect of excluding the morphological information since the Wiktionary data does not appear to be sensitive to it. This resulted in a lower WER. The improvement can be mostly explained by the appearance of the aforementioned common errors, such as the incorrect digraphs and hiatus filling.

### 4.2 Speech synthesis

We used the Tacotron 2 system (Wang et al., 2017) to extrinsically evaluate our model. The Tacotron 2 model is a sequence-to-sequence model trained to map a sequence of letters to a sequence of features that encode the audio. These features are audio spectrograms computed on short segments of audio thus they capture not only the pronunciation, but also the volume, speed and intonation. This representation is then converted to waveform using a WaveNet-like architecture (van den Oord et al., 2016).

The lack of freely-accessible Hungarian speech data has limited our options, thus we have settled with the Hungarian part of the CSS10 dataset[7] created by Park and Mulc (2019). It is essentially the annotated and segmented audiobook of *Egri Csillagok* 'Eclipse of the Crescent Moon'[8] read by a single speaker.

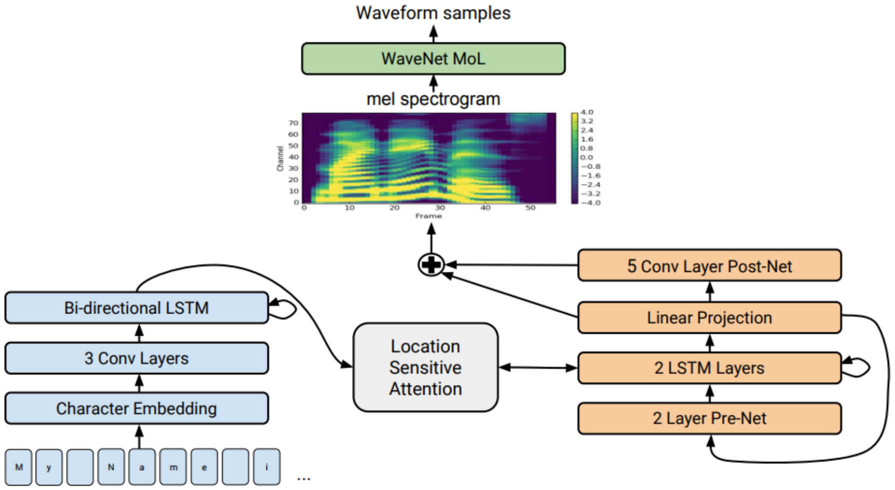

Fig. 1: Architecture of Tacotron

In our test setup, we used the Tacotron implementation of Nekvinda and Dušek (2020)[9]. We trained the Tacotrons for 300 epochs with the default hyperparameters presented in the Github repository, and we have used their pre-trained model from the Github repository of their WaveRNN[10] implementation. We compare the models that were trained on the original text to the model trained on the phonetically transcribed text. The training of both models took around 1 day on a single 1080Ti GPU.

Initially, we expected the two models to be almost indistinguishable, since the labels (audio recordings) are the same, and the difference between the training datasets are not that significant. However, our impressions of the difference is that it is perceptible, even by the naïve ears, but this distinction is not easy to quantify.

The baseline model can already produce clean, natural, comprehensible speech, and our goal was to achieve similar performance to prove that our transcriber

[7] `https://www.kaggle.com/bryanpark/hungarian-single-speaker-speech-dataset`

[8] `https://en.wikipedia.org/wiki/Eclipse_of_the_Crescent_Moon`

[9] `https://github.com/Tomiinek/Multilingual_Text_to_Speech`

[10] `https://github.com/Tomiinek/Multilingual_Text_to_Speech/releases/download/v1.0/wavernn_weight.pyt`

does not lose information. From this point of view, we succeeded. Our model, which trained on the phonetic transcription, is by no means inferior to the baseline model. There is a subtle difference in intonation and surprisingly, in tone, but one of the starker differences was audible in case of /h/-s and affricates. The /h/-s are more pronounced and somewhat more natural in our model, which is expected since they are generally rare in occurrence but are influenced by many rules. On the other hand, our model makes less of a distinction between affricates and fricatives. In many cases, /t͡s/ and /s/ sound too similar. This can happen due to our tool merging too many *-tsz-* and *-dsz-* segments, when in reality, these are frequently pronounced differently from *-c-*.

## 5 Conclusion and further research

In this paper, we have presented an automatic phonetic transcriber tool which is publicly available, fast, and fits well the existing emtsv framework, thus it is easy to use. The small amount of publicly available data limited our ability to fully evaluate the model, but as we have shown in Section 4, the transcriber performs well.

There are multiple directions for further research. On one side, the transcriber's accuracy can easily be improved with the usage of a curated pronunciation lexicon. As another direction, the speech synthesis models have shown that the transcriber's performance can be 'heard', giving us an interesting insight into what happens when we force the system to learn based on the phonetic forms. We are certain that there are more interesting phenomena happening here which could be discovered by experienced phoneticians.

## 6 Acknowledgements

This work was supported by the Lendület Research Grant of the Hungarian Academy of Sciences (Lendület 96233 'Profiling learning mechanisms and learners: individual differences from impairments to excellence in statistical learning and in language acquisition', PI: Ágnes Lukács). The work of Lévai is partially supported by NRDI Forefront Research Excellence Program KKP_20 Nr. 133921, Hungarian National Excellence Grant 2018-1.2.1-NKP-00008, and by the Ministry of Innovation and the National Research, Development and Innovation Office within the framework of the Artificial Intelligence National Laboratory Programme.

## Bibliography

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale (2020)

Gósy, M.: Fonetika, a beszéd tudománya. Osiris Kiadó, Budapest, Magyarország (2004)

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Kundráth, P., Vadász, N.: `emtsv` — egy formátum mind felett. In: Berend, G., Gosztolya, G., Vincze, V. (eds.) XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019). pp. 235–247. Szegedi Tudományegyetem Informatikai Tanszékcsoport, Szeged (2019)

Nekvinda, T., Dušek, O.: One model, many languages: Meta-learning for multilingual text-to-speech (2020)

Nemeskey, D.M.: Natural Language Processing Methods for Language Modeling. Ph.D. thesis, Eötvös Loránd University (2020)

Novák, A., Siklósi, B.: Grapheme-to-phoneme transcription in hungarian. Int. J. Comput. Linguistics Appl. 7, 161–173 (2016)

Novák, A., Siklósi, B., Oravecz, C.: A new integrated open-source morphological analyzer for Hungarian. In: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (may 2016)

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. CoRR abs/1609.03499 (2016), http://arxiv.org/abs/1609.03499

Park, K., Mulc, T.: CSS10: A collection of single speaker speech datasets for 10 languages. CoRR abs/1903.11269 (2019), http://arxiv.org/abs/1903.11269

Siptár, P., Törkenczy, M.: The phonology of Hungarian. Oxford: Oxford University Press. Phonology 18(2) (2000)

Szende, T.: Illustrations of the IPA: Hungarian. Journal of International Phonetic Association pp. 91–94 (1994)

Wang, Y., Skerry-Ryan, R.J., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q.V., Agiomyrgiannakis, Y., Clark, R., Saurous, R.A.: Tacotron: A fully end-to-end text-to-speech synthesis model. CoRR abs/1703.10135 (2017), http://arxiv.org/abs/1703.10135

Yolchuyeva, S., Németh, G., Gyires-Tóth, B.: Transformer based grapheme-to-phoneme conversion. Interspeech 2019 (Sep 2019), http://dx.doi.org/10.21437/Interspeech.2019-1954