2025 ACTA UNIVERSITATIS CAROLINAE PAG. 43–60
PHILOLOGICA 3

CUSTOMISING CZECH PHONETIC ALIGNMENT USING HuBERT AND MANUAL SEGMENTATION

ADLÉTA HANŽLOVÁ, VÁCLAV HANŽL

ABSTRACT
This paper presents Prak, a forced alignment tool developed for Czech, with a focus on transparent modular design and phonetic accuracy. In addition to a rule-based pronunciation module and exception handling, Prak introduces a novel application of non-deterministic, backward-processing FSTs to model complex regressive assimilation processes in Czech consonant clusters. We further describe the integration of a HuBERT-based transformer model and training including extensive manually time-aligned data to enhance phone classification accuracy while maintaining ease of installation and use. Evaluation against a manually aligned test corpus demonstrates that the enhanced model significantly outperforms both our earlier Prak-CV model and the long-established previous forced alignment baseline. The new model reduces major boundary errors and mismatches, bringing alignment accuracy closer to manual phonetic segmentation standards for Czech. We emphasize both methodological transparency and practical usability, aiming to support phoneticians working with Czech as well as developers interested in extending the tool for other languages.

Keywords: forced alignment; phonetic segmentation; Czech; HuBERT; Prak; Praat

1. Introduction

The majority of phonetic analyses require labelling recordings to identify their contents (the type of content varying based on the research question) in order to accurately measure properties that are to be related to said contents. Among the most common types of audio labelling is identifying phone boundaries as a means to then measure formant frequencies, spectral or temporal properties, assess pronunciation and much more. Labelling recordings is often a very tedious and time-consuming process, but also one that is vital to ensure the validity of measurements that are made based on the determined time boundaries. The general effort therefore is to automate these processes as much as possible using forced alignment software tools.

There are many forced alignment software tools available, most of them focusing on phone alignment of English-language material (for a comprehensive list see Pettarin, 2018). Only a small subset of these support multiple languages and even the ones that

https://doi.org/10.14712/24646830.2025.20

do often do not include Czech as an option. Among the most used alignment tools with support for multiple languages generally employed in phonetic research are the Munich alignment system MAUS (Schiel, 1999) with its web-based implementation, and the Montreal forced alignment software (McAuliffe et al., 2017). Tools developed specifically for alignment of Czech-language material include Prague Labeller (Pollák et al., 2005, 2007) and more recently, Labtool (Patc et al., 2015) and Prak (Hanžl & Hanžlová, 2023). There is also a forced alignment tool directly integrated in Praat (Boersma & Weenink, 2023), which is useful as an easy-to-access tool. It has, however, potential for further upgrading and development (personal conversation with Paul Boersma, 2023). Some features of these tools will be elaborated on below.

When editing a sound and textgrid simultaneously in Praat's *View & Edit* window, selecting the menu item *Interval / Align Interval* (or Ctrl+D) will add a word and phone tier to the existing textgrid with time-aligned intervals. The option works for many languages (including Czech) and can align single words or short phrases. The alignment tool in Praat uses a speech synthesizer to create an audio track based on the provided orthography. The created sound is then compared to the provided audio and aligned using dynamic time warping (Boersma et al., 2023). This makes the alignment option simple to implement, but also limits its use when aligning longer sequences, especially ones containing pauses, as these are not reliably identified by the algorithm. The option also works exclusively from the *View & Edit* window and doesn't have a setting for automatic alignment of multiple files.

A very widely used forced alignment software with easy access and no installation is the Munich Automatic Segmentation System (MAUS, Schiel, 1999) with its web service (Kisler et al., 2017). The use of the MAUS web service is free for members of academic institutions for non-profit use, otherwise users must obtain a license to use it (Bavarian Archive for Speech Signals, 2018). The alignment software supports over 30 languages, with multiple dialect variants for English and German. The list of supported languages does not, however, include Czech. The web-based interface makes forced alignment using MAUS easily accessible with only a web browser and internet connection, but obscures the source code and does not therefore allow tweaking the way the system runs or implementation of the user's own models, such as models trained for other languages.

The forced alignment software that was until recently very commonly used to align Czech recordings for purposes of phonetic research is Prague Labeller (Pollák et al., 2005, 2007), developed at the Czech Technical University. The aligner uses HTK GMM models and employs a rule-based pronunciation generator. The software was a state-of-the-art forced alignment tool at the time of its development and was in consistent use for more than a decade at the Institute of Phonetics, Charles University. We have included a comparison of this aligner with our newly developed tool in Section 5. Development of new language model software options in the recent years has led to the tool becoming impractical to install with its dependencies. There is also a demand for higher accuracy of the automatic alignment. Additionally, the software is not freely distributed and runs only under Windows, which prevents the public, including students of phonetics, from using the alignment software with their own devices.

There has also been an effort to develop a newer forced alignment tool focused on Czech, implementing HMM-based phonetic segmentation using Kaldi instead of HTK models (Patc et al., 2015). The main focus of this tool was to enable detailed study of pronunciation variation in spontaneous Czech speech through automated segmentation and variant detection, integrating Kaldi's acoustic modelling techniques. Experimental results showed that Kaldi-based models provided more consistent and precise phone boundaries compared to older HTK-based methods. Despite these results, the software is not in public distribution and has not been widely deployed by Czech phoneticians.

A similarly recent Kaldi-based forced alignment software with a wide range of supported languages, including Czech, is the Montreal Forced Aligner (MFA, McAuliffe et al., 2017), which is available under a MIT license (Opensource.org, 2025). The installation of the MFA requires Kaldi as a dependency and the download of models for the desired language. The aligner includes several pre-trained models for Czech, the more basic ones being trained on the same CommonVoice (Ardila et al., 2019) database as the original model in Prak (Hanžl & Hanžlová, 2023). More advanced models use training data from larger databases, including paid datasets. The option to choose from multiple models allows some customization for the user, nevertheless, the phone sets implemented in the models for Czech may not be sufficient for the purposes of detailed phonetic research (as discussed in Hanžl, 2023). The MFA tool does allow for implementation of the user's own models if so desired, so these issues can be resolved, but the installation is still quite convoluted, so developing a new easy-to-install tool along with more precise models is a logical step in the process of ameliorating Czech forced alignment options.

The most recent tool which specifically aims to provide a streamlined installation process as well as resolve known automatic segmentation issues and improve phone alignment of Czech recordings (with the option to train and implement models for other languages) is Prak (Hanžl & Hanžlová, 2023). Similarly to the MFA, this software is freely available on all major computer operating systems under a MIT license. However, unlike most forced alignment tools that are in wide use, Prak requires only minimal dependencies and aims to keep its architecture simple in order to not only be usable as a user-friendly alignment tool, but also to enable other researchers or programmers in the future to build on it without restrictions. The default model provided in the free distribution of Prak is trained on CommonVoice (CV) Czech recordings, as mentioned above. Due to its simplicity, the tool is useful even for non-phoneticians, such as students in a pronunciation class, to help navigation in longer sound files by quickly obtaining an overview of the contents of a recording.

While the aligner with this CV-trained model significantly outperforms the forced alignment softwares for Czech that have been in use before, the phone boundaries still often need to be moved manually after the automatic alignment to achieve the precision needed in phonetic research. The newest step in the development of Prak therefore was to train a new model in collaboration with the Institute of Phonetics, Charles University, which would use manually aligned recordings and HuBERT (Hsu et al., 2021) embeddings in addition to the original training dataset in order to more closely mimic human behavior based on established Czech segmentation rules (Machač & Skarnitzl, 2009), hopefully further reducing the amount of manual labor necessary after using the forced alignment software.

In this paper, we aim to present Prak and its functionality in a comprehensive way, as well as provide detail about the training of the new model and compare both Prak models with the output from Prague Labeller as the long-standing predecessor in Czech phonetic alignment. Our goal is twofold: first, to introduce our software to its intended users; and second, to present the underlying concepts and development strategies in sufficient detail to enable future developers and researchers to build upon it. To that end, this paper is structured to reflect both the practical and methodological dimensions of the tool.

We provide an outline of the installation process, including software prerequisites and integration with the Praat environment. This is followed by a description of the pronunciation modeling framework, including built-in replacement rules, the exceptions file, and the modular Finite State Pronunciation Blocks. We then detail the training procedure for the HuBERT-based model, discussing both the use of additional manually aligned data and the architectural considerations that informed our design choices. Finally, we evaluate the performance of the system, comparing both Prak models with the Prague Labeller using manually aligned data as a reference, and report on alignment accuracy in terms of phone identity and boundary placement.

2. Design and installation

2.1 Installation of the software and prerequisites

The use of Prak requires only two external software tools, namely Python 3 (Van Rossum & Drake, 2009) with the PyTorch (Paszke et al., 2019) and TorchAudio (Yang et al., 2022) libraries. The installation of these prerequisites is clearly outlined in the official Prak installation instructions (Hanžl & Hanžlová, 2025), and no knowledge of programming or speech technology is necessary to complete the setup. The documentation provides step-by-step commands that can be run via command line, and offers platform-specific guidance where relevant. This makes the software accessible to phoneticians and linguists as well as other researchers or students who may not necessarily have a technical background. At the same time, this simple and modular structure ensures that the code remains easily readable and modifiable for programmers or developers who wish to extend its functionality.

The Prak installation process further involves only downloading the Prak code (via Github or as a zip file) and choosing the desired model for alignment. Available options include the basic Prak-CV model (as presented in Hanžl & Hanžlová, 2023) or the more fine-tuned, recent model based on HuBERT (Hsu et al., 2021). Details regarding the HuBERT-based model and its properties are provided in Section 4. Once installed, Prak can be run via command line or through a script which integrates Prak's functionality into the GUI in Praat (Boersma & Weenink, 2023) while also adding supplementary features, as described in the following section.

2.2 Praat GUI integration and additional functionality

Apart from functioning directly from the command line, Prak provides a Praat script which embeds the main Python forced alignment tool and can be added to Praat's dynamic menu for easy access. The script also performs several assessments of the input files and provides additional options for the alignment. First, basic file checks are performed. The number of sound and textgrid files are counted and compared to ensure all desired recordings have a text input to be used in the alignment. The Praat interface also provides an additional option to use only one text input to align multiple sound files. Sound and textgrid names are also compared and in case of name mismatch, the user is prompted to decide whether the combination of the files with different names was deliberate. When working with multiple files, this check can be overruled and sounds aligned by the order in which the items are open in Praat.

The contents of each textgrid provided to the tool are also reviewed in order to determine the source text correctly. The tool expects the tier containing the source text to be named "phrase" and outputs three tiers after performing the alignment: a "phone" tier containing the phone boundaries, a "word" tier containing word boundaries and a "phrase" tier with the original source text. This is modelled after the Prague Labeller (Pollák et al., 2005, 2007) output, established as the standard at the Institute of Phonetics, Charles University. An example of the output textgrid is presented in Figure 1. The script integrating Prak into the Praat UI firstly checks that the source textgrid doesn't contain a non-empty "phone" tier to prevent accidental overwriting of files. If such a tier is found in the textgrid, the user is notified of this circumstance and can choose to either stop the script or continue and overwrite said file. Similarly, if a "phrase" tier is missing in the source textgrid, the user is prompted to identify the tier containing the source text which is to be used.

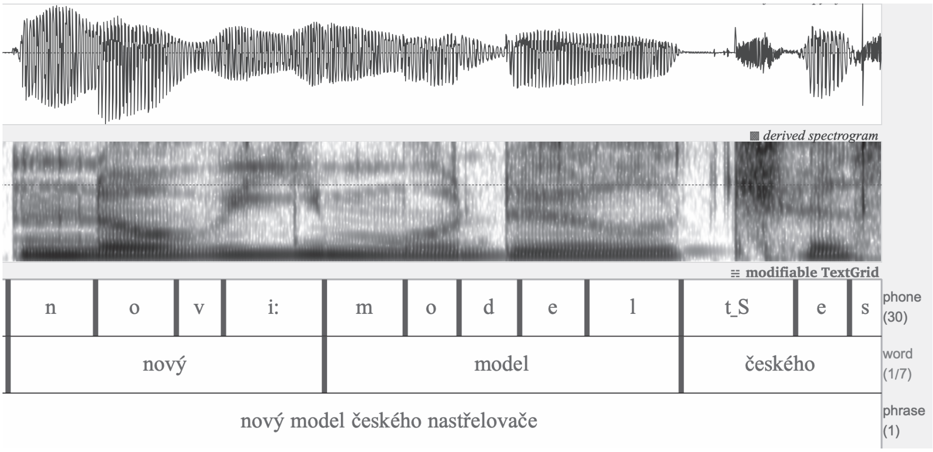

Figure 1 Example of an output textgrid after forced alignment using Prak.

If all assessments of source files are successful, the sounds and their corresponding text sources are fed to the Python tool to proceed with forced alignment. In the alignment process, textgrids containing the source text are replaced with output files containing the three tiers described above. All other tiers that may be in the original textgrids in addition to the source text are ignored and are not part of the final aligned files. All output textgrids are also renamed after the sound files they correspond to.

3. Pronunciation rules

The pronunciation generator in the original Prak used several novel principles, trying to remedy deficiencies of pronunciation generators used in previous decades, especially trying to make the set of rules manageable long term and the level of detail adjustable should the scientific research at hand require such a change. An overview of the phonetic alphabet used, text cleanup issues and basic design of the pronunciation generator was presented in our original Prak introduction publication (Hanžl & Hanžlová, 2023). While we mostly reused the generator unchanged for the new version of Prak, the available descriptions of the design principles and implementation are rather superficial, leaving direct inspection of the source code as the sole option for researchers seeking insight into the details. We therefore take the opportunity to describe these components in a format that is more accessible and comprehensible to readers.

The primary user group targeted by the design of Prak are researchers who are likely to fine-tune the pronunciation rules logic. Simple dictionary-based approaches often employed for English are largely insufficient for Czech, which is a highly flexible language, necessitating many entries for all the forms of every word and making manual ad-hoc additions of new words to the dictionary quite cumbersome. Usual approaches based on replacement rules are also hard to apply, mainly due to large consonant clusters with complex assimilation rules in Czech pronunciation, where, among other processes, regressive assimilation of voicing applies to all viable consonants within said cluster (Skarnitzl, 2011, p. 123). Coping with the voicing or devoicing of phones presents a considerable challenge. As a result, the scope of our pronunciation generator tool is rather narrow, addressing the specific needs of phoneticians working with the Czech language. Nonetheless, there is potential for reusing components of our software in other languages with similar phonotactic and assimilation patterns, such as Polish, Slovak, Russian, or even Armenian (Kuldanová et al., 2022; Pavlík, 2009).

After decades of experience with the replacement-rule-based Czech pronunciation generator used in Prague Labeller (Pollák et al., 2005, 2007), we decided to address the main known shortcoming: The rules table grew to hundreds of entries over years of use, and while this approach worked, inserting a new rule in the correct position among existing ones became a highly expert task, as it always required verification on a large corpus of previously generated pronunciations, identifying all changes caused by the new rule, and determining whether they served the intended purpose. Making pronunciation rules position-independent was therefore an important design goal of Prak. This initially seemed difficult, as the rule order also corresponded to gradually changing layers of representation which started with graphemes and gradually progressed through phones to allophones. However, we were able to find a practical solution, structuring the pronunciation processing in two layers using two different approaches:

1. A set of replacement rules without any human-defined order. The rule with the longest match is applied first. The rest of the word is then subject to more possible replacements but whichever part was already touched by another replacement rule is not affected by any other. This part can be easily used to specify pronunciation of "foreign looking" substrings and stay close to the graphemes.

2. A Finite State Machine based layer mainly taking care of Czech assimilations. This layer can be adapted to a particular research goal and the corresponding details of phonemic representation but does not require changes as new speech material is being added.

3.1 Built-in replacement rules and Exceptions file

Built-in rules deal with the most common patterns of foreign pronunciation in Czech. They can also serve as a didactic example for entries in the Exceptions file, as the format of both is the same and in practice, they are mixed by Prak into a single optimized structure with priority assigned by match length. Any built-in replacement rule can therefore be overridden in the Exceptions file simply by using a longer (more specific) string to be replaced. A notable feature of the replacement rule engine in Prak is its ability to consider multiple replacements. Each rule specifies a substring to be found in a word and lists one or more possible replacement strings. Selection of the right pronunciation version is later determined by the acoustic properties of the signal being processed.

As mentioned above, in addition to the built-in replacement rules, the Prak source contains an Exceptions file, where further pronunciation rules can be specified by the user. The built-in pronunciation generator is very meticulous in considering possible assimilations (of various kinds) and even glottal stop presence, which is not always required by the orthoepic norm (Volín, 2012; Volín & Skarnitzl, 2018, p. 22) and can have multiple acoustic realizations (Machač & Skarnitzl, 2009, pp. 125–131). However, due to the nature of the pronunciation generator, as described below, Prak has limited lexical knowledge and therefore may require additional input for handling cases that fall outside the scope of general pronunciation rules.

The Exceptions file is consulted when Prak is invoked from Praat. It uses very simple entries, modelled after the built-in rules, which are easy to follow and add to as the need arises. The file allows users to manually specify the pronunciation of strings at or below word-level which then override any other rule that may otherwise be applied to said strings within the default processing. This is particularly useful for dealing with proper names, loanwords, abbreviations, or unusual morphophonological irregularities that are difficult to capture systematically. Each entry in the exceptions file maps a written string directly to its target phonetic or allophonic representation, ensuring accurate alignment in contexts where automatic rule application could be unreliable. Specific instructions along with examples of added pronunciation rules can be found on the Prak installation page linked in Section 6.1.

3.2 Finite State Pronunciation Blocks

Finite State Transducers (FSTs) have appeared as a unifying concept in some speech recognition systems in the past. However, using backward-going FSTs to translate one symbol sequence into another as a pronunciation assimilation tool is a rather unique feature of Prak, and we believe it is a highly efficient option for Czech (and potentially for other structurally similar languages). Furthermore, we use non-deterministic FSTs, which may suggest multiple output symbol options. The convolution of several such non-deterministic FSTs, together with the potentially multi-option replacement rules described above, creates a very powerful tool capable of handling complex Czech assimilations, even in words of foreign origin. At the same time, describing possible pronunciations remains mentally manageable, as it is decomposed into clearly understandable parts – rules handling foreign elements at a level close to graphemes, and FSTs dealing with assimilations, with each simple FST addressing just one phenomenon. The resulting pronunciation options can then be visualized as a so-called "sausage" structure, offering a more accessible alternative to Directed Acyclic Graphs.

The need for the use of FSTs arises from ongoing efforts to enumerate the possible assimilation changes in Czech consonant clusters. As mentioned in Section 3.1, in the Czech language, many of these changes are like a domino effect going backward in a sequence of possible phones. The change can be rather far reaching, and consonant clusters can be remarkably long. For example, in the approximately 6,000 different words in our training set, nearly 700 distinct consonant clusters were identified, 17 of which had a length of five or more characters. Capturing the essence of assimilation logic in FSTs turned out to be a practical solution to dealing with this vast variety.

Figure 2 demonstrates a FST handling backward assimilation of the voiced/unvoiced property. The current FST state depicts the influence being exerted on the phone to the left. For clarity, only a subset of the edges is shown in the figure (edge labels are in the format Input/Output). The word "kdyby" is processed right to left, changing the consonant cluster "kd" to [gd].

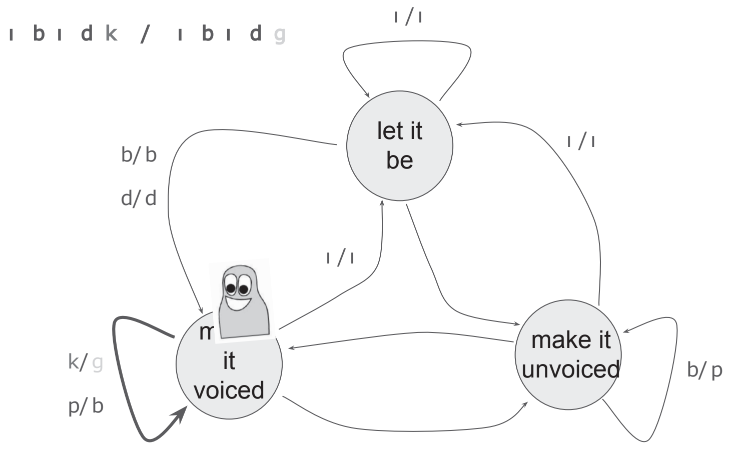

Figure 2 Illustration of regressive assimilation of voicing, as processed through backward traversal of the voiced/unvoiced property by a FST. Token (pawn) represents the current state.

The current state is depicted as a token (pawn), which can travel along edges, consuming an input symbol I and producing an output symbol O when moving along an edge labeled I/O. Should there be multiple possible edges with the same matching input symbol I, the token "clones" itself into multiple tokens, each following a different path and creating separate branches in the pronunciation "sausage" graph. When clones meet at the same state, the tokens merge again along with the pronunciation branches they represent.

A typical example of this branching occurs at word boundaries. Depending on the degree of word separation, the assimilation domino effect may either cross the boundary or stop at it. Another example – modeled by a different FST – would be pronunciation of the word “galantní”, where the cluster “ntn” can be realized as [ntɲ], [ncɲ], or [ɲcɲ]. This represents another case of non-deterministic regressive assimilation that can be effectively modeled by FSTs. The actual Prak algorithm is somewhat more nuanced than simply applying non-deterministic replacement rules followed by a convolution of several backward-going non-deterministic FSTs, but a large portion of Czech pronunciation logic can indeed be captured using this scheme. Further details regarding the pronunciation generator and its implementation can be observed in the Prak source code.

4. Training the new HuBERT model

The original Prak release prioritized simplicity, easy maintainability of the codebase and simple installation. The neural network architecture of the Prak-CV model used only a very simple stack of ReLU (Rectifying Linear Unit) layers for phone classification. The goal was to design the system in a way that would be accessible to researchers who are mainly interested in phonetics, rather than artificial intelligence specialists who are more likely to experiment with complex and rapidly evolving architectures. Even under this restriction, the improvement in precision was substantial when compared to the previously employed Gaussian mean models from the HTK toolkit era. Nonetheless, transformer-based embeddings have become so widespread in recent years (Lin et al., 2022) that we decided to incorporate them in an improved version of our phone classification module.

Maintaining relative simplicity was still among our aims for the new model, therefore, we selected a freely available neural network model that computes HuBERT embeddings (Hsu et al., 2021). This network is pre-trained on a mix of many languages and coefficients of this neural network are available for automatic download from public servers at the time of first inference on a particular computer. Even though it would be possible to fine-tune coefficients of the HuBERT network, doing so would compromise the simplicity of installation and coefficients of the fine-tuned HuBERT network would need to be included in the distribution of Prak itself, making the distribution package many times bigger. Instead, we decided to use the HuBERT network in its default form and train an alternative of our original simple ReLU stack which would use HuBERT embeddings as an additional input. This approach proved to be largely sufficient for our goal of achieving more precise identification of phone identities and time boundaries.

4.1 Using additional manually time-aligned training data

As mentioned in the introduction, the original Prak phone classifier model was trained exclusively on the freely available CommonVoice dataset (Ardila et al., 2019). The goal was to enable easy portability to other languages – the CommonVoice database is available for many languages, not only Czech – as well as keep the prospect of future precision improvements with retraining on the ever growing CommonVoice data. All phone boundaries in the original model were automatically derived during training, as the CV dataset contains only recordings of sentences and their orthographic transcripts. Any exact match between manually annotated and automatically derived phone boundaries is therefore just a result of a coincidental match between human judgment and the system's estimation of the most likely boundary location (nevertheless, this match is often notably close).

The match achieved by the original Prak model was still a significant improvement over the tools used previously – this fact being a tribute not only to Prak but also to the human team working on the reference labeling, trying to make evidence based decisions based on an agreed upon sensible set of rules. Convergence of the time boundaries derived by these two independent processes, one based on human expertise and manual annotation and one purely data driven, was remarkable. However, greater precision was still achievable, and the long-term goal remained to train a model which would place time boundaries where human researchers expect them, given their specific research approaches and objectives. In line with this goal, we added data with manually time-aligned boundaries to the training dataset for the new HuBERT-based Prak model.

The dataset used for the training of this model in addition to the CV recordings used in the original model was a corpus of manually corrected time-aligned recordings, generously provided by the Institute of Phonetics, Charles University. This additional dataset consists of 1435 recordings with a total duration of 5 hours and 15 minutes. All recordings in the dataset were aligned by a forced alignment software (the majority by Prague Labeller) and subsequently manually checked and adjusted to comply with the Czech standards for phonetic segmentation (Machač & Skarnitzl, 2009)[6]. The manual alignment was done for the purposes of conducting phonetic research (see for example the research by Volín & Skarnitzl, 2022, which uses a subset of this corpus) rather than training a model for forced alignment software. This corpus should therefore reflect the needs of Czech phoneticians in terms of the standards expected when conducting research. The exact contents of our time-aligned training dataset are presented in Table 1.

Table 1 Overview of the contents of the corpus of manually corrected time-aligned recordings used in addition to the CommonVoice dataset in training the new Prak model.

type of text	type of speaker	total duration	n female speakers	n male speakers
audiobooks	professional	134 m 10 s	6	5
poetry reading	amateur	89 m 22 s	14	12
radio news	professional	59 m 25 s	5	11
	amateur	31 m 49 s	8	1
total		314 m 46 s	33	29

The dataset consists of recordings from three different genres, recorded under different conditions by both professional and amateur speakers. The first genre, accounting for approximately 2.25 hours of the recordings, were spoken narratives (i.e. storytelling)

[6] Based on our approximation, the labelling of this corpus of recordings took more than 300 hours of manual labour in addition to using a forced alignment software.

extracted from audiobooks recorded by experienced actors in professional studios and produced by renowned publishers. The extracts used in the training dataset were read by 6 female and 5 male speakers.

The second type of recordings in our training dataset were readings of poetry, the total duration of which was around 1.5 hours. These samples of poetry reciting were recorded by 14 female and 12 male speakers. The recordings were done in the sound treated studio of the Institute of Phonetics in Prague. The speakers were volunteering students of philology with an interest in poetry.

The third genre included in the dataset were two types of news reading with a total duration around 1.5 hours, similarly to the poetry reciting samples. Two thirds of this subset (around 1 hour) consist of recordings of authentic news-bulletins from Czech radio broadcasts, recorded by 16 (5 female, 11 male) professional speakers. The remaining third of the samples are texts taken from said Czech radio broadcasts read by 9 volunteering students (i.e. nonprofessional speakers; 8 female, 1 male) in the same studio the poetry reciting was recorded.

4.2 Details of the HuBERT model

The HuBERT model computes a transformer-based embedding, similar in principle to word representations used in many translation systems and artificial intelligence dialogue systems, and analogous to amino-acid context-aware representations in modern approaches to analyzing or even synthesizing proteins in biochemistry, among other uses. The transformer-based processing has become a highly successful overarching paradigm (Lin et al., 2022; Vaswani et al., 2017). HuBERT, in particular, applies this paradigm to short (20 ms) segments of the speech signal, representing these as long vectors of numbers describing not only the spectral characteristics of the specific sound (as traditional cepstral features do) but also the meaning of the sound chunk in a particular context.

The HuBERT model is pre-trained on a large multilingual corpus of unlabeled speech. Similar to training procedures in other domains, this process involves masking short segments of the speech signal and requiring the model to reconstruct these masked parts as precisely as possible. This forces the model to capture long-range dependencies in the speech signal, enabling it to perform speaker adaptation and other phone-level analyses, ultimately acquiring enough knowledge to fill in the missing segments with high confidence (Boigne, 2021). In practical use of the pre-trained HuBERT model, our goal is often different, as we typically have the complete recording without any missing segments. We instead leverage the internal representation developed during the training and reuse it for particular tasks at hand. As it turns out, these internal representations are very rich and context-aware and can be applied to a variety of tasks with notable success.

There are two ways to use the pre-trained transformer-based models:

1. Keep the pre-trained model unchanged and train an additional neural network connected to it. The added network uses the pre-trained representations as inputs and is trained to produce the desired outputs.
2. Train not only the additional network but also fine-tune the pre-trained model itself.

We decided to employ the first approach, which provided satisfactory results for our purposes, so we kept it as the final solution. As mentioned above, while the second approach could theoretically achieve even better results in our tests, it would also come with the technical disadvantage of having to distribute the modified HuBERT parameters together with Prak, making the installation package significantly larger. Apart from this technical nuisance, there is also another risk: further training of the HuBERT coefficients could actually reduce Prak's precision in practice due to overfitting (see Chicco, 2017), as our training datasets are quite small for a model of HuBERT's size. Given all these considerations, we opted to use the unchanged HuBERT coefficients.

There are multiple versions of the HuBERT model – the BASE version, pre-trained on 960 hours of unlabeled audio from the LibriSpeech dataset (Panayotov et al., 2015), and the larger LARGE and XLARGE models, pre-trained on 60,000 hours of speech. Given the relative simplicity of our phone alignment task – compared to other HuBERT applications such as speech recognition – and our overall goal of maintaining simplicity, we opted for the smallest BASE model (Torchaudio Contributors, 2024).

The original Prak used a 10 ms time resolution for its cepstral features, while HuBERT uses 20 ms time steps. We certainly did not want to make Prak-detected time boundaries more coarse-grained (on the contrary, we would even consider a 5 ms time step for future work), so we simply repeated each HuBERT output vector twice. This way, HuBERT contributes mainly the long-term contextual information, while the 10 ms cepstral features allow for a sharper local resolution.

The HuBERT model uses a multilayer transformer input stack, computing different embeddings at each level. Each additional layer of this stack should theoretically produce increasingly more abstract and more context-aware embeddings, making the top layer the most information rich. In practice, any layer can be used as input for the subsequent processing, not restrained to the top one. Using one of the lower layers offers the potential benefit of computational savings, as the upper layers do not have to be computed. We therefore trained several variants of the system, aiming to identify the lowest layer that still allows the system to operate with negligible precision loss compared to the best (likely the topmost) transformer stack layer being used. After multiple training attempts, we selected layer 7, though the differences in performance across layers were relatively minor.

HuBERT BASE uses a 16 kHz waveform as input (the same as the original Prak). Instead of MFCC, the pre-trained HuBERT model uses a 7-layer Convolutional Neural Network which does a learned feature extraction, generating a 512-dimensional vector every 20 ms. These vectors are then processed by a stack of 12 transformer layers, each using 12 attention heads. Each 20 ms unit is represented as a vector of 768 numbers when passing between layers. This is the representation that is forked to the Prak phone classifier network.

The original Prak model used 13-dimensional MFCC features with 9 context frames on both the left and right, resulting in (9 + 1 + 9) × 13 = 247 numbers, further augmented by 4 × 13 = 52 speaker adaptation values, for a total of 247 + 52 = 299 values every 10 ms. The HuBERT-based Prak model retains this entire representation and concatenates it with the 768-dimensional vectors from HuBERT's 7th transformer layer (each vector being used twice, as HuBERT operates with 20 ms chunks), resulting in 299 + 768 = 1067 values every 10 ms. This representation is fed to the 3-layer ReLU stack with an internal vector size 100, followed by a final softmax layer to produce phone probabilities.

Among the phone probabilities is also the probability of a «silence» sound which includes not only a real silence but also all non-speech events like breaths or hesitations, as encountered in the training data. Common Viterbi decoder then decides globally optimal phone identities and silence boundaries, given the choices proposed by a pronunciation generator.

5. Evaluation and comparison of phone alignment

5.1 Method and material

In order to evaluate the performance of our new HuBERT-based model, we compared the output of Prak alignment using this model with both the previous Prak-CV model (Hanžl & Hanžlová, 2023) and the Prague Labeller (Pollák et al., 2005, 2007), using manually aligned recordings provided by the Institute of Phonetics as the ground-truth evaluation baseline. We measured the percentage of phone identity mismatches compared to the baseline, as well as the percentage of phone boundary misalignments.

We contrasted the generated pronunciations, counting phone insertions, deletions and substitutions. Direct comparison of phone identities determined by the forced alignment tools is not straightforward, as the phone sets differ based on the inventories used by each aligner. For instance, compared to the Prague Labeller, Prak also detects glottal stops, distinguishes between voiced and voiceless "ř" or accounts for assimilations at word boundaries. Manually time-aligned textgrids may, on the other hand, reflect slightly different approaches to phone identity labeling, depending on the research questions for which the recordings were originally intended. Additional annotations of some segments, such as syllabic [l̩] or [r̩], may also be present and contribute to the phone identity mismatch percentage.

At places where phone identity matched, we measured time shifts of the phone boundaries relative to the manual reference. While the phone identity may in some cases be subject to interpretation, as discussed above, the time positions in the manually aligned reference data should be in compliance with the segmentation standard for Czech (Machač & Skarnitzl, 2009) and can therefore be considered accurate for the purposes of Czech phonetic research. An important parameter affecting the efficiency of manual correction of automatically aligned files is the frequency of boundary misalignments that require adjusting multiple phone boundaries to correct the error. We therefore used the number of boundary shifts exceeding thresholds of 100 and 200 ms as a quality measure of major boundary misplacement.

For our evaluation, we used a subset of the phonetic corpus presented in Section 4.1. We selected 156 recordings (balanced across the file subtypes in the dataset) that were not used at any stage of training for our HuBERT-based model or any other model. These recordings have a total duration of approximately 30 minutes and contain about 20,000 individual phones. We obtained the original output TextGrids of forced alignment produced by the Prague Labeller without any manual correction, and we performed forced alignment of the same recordings using Prak with both the CV and HuBERT models. The source text was copied from the manually labeled files and used as input in all three alignment iterations.

When we examined individual cases of divergence between manual labeling and Prak-HuBERT labeling, we found that most differences were caused by non-uniform manual labeling, for example, the use of different phone sets, the omission of certain phenomena, or, conversely, the inclusion of additional detail. While we invested substantial effort in automatic normalization of all the manually labeled data to a common standard, this process had inherent limitations. To gain additional insight, we further double-checked the manual labeling in our test set for errors and compliance with the selected transcription method in cases where it diverged from the Prak-HuBERT labelling. We then re-evaluated the Prak-HuBERT model and observed, for instance, nearly ten times fewer boundary shifts exceeding 0.1 s. We therefore added an additional test for shifts over 50 ms to gain finer granularity. While the results in this additional table are truly impressive, they are no longer strictly objective and should be interpreted as suggesting that Prak-HuBERT errors are approaching the limits of what can be reliably measured.

5.2 Results and discussion

Table 2 shows the percentage of errors in the phone identity mismatch and boundary misplacement tests, as well as the cumulative results of these tests, comparing the output of Prak's models and the Prague Labeller with data from manually time-aligned files. It is evident that Prak generally outperforms its predecessor in all types of tests provided, indicating a significant decrease in errors leading to manual corrections requiring adjustment of more than one boundary.

Table 2 Percentage of phone mismatch and boundary misplacement, comparing the output of Prak's two models and the previously most used forced alignment tool with manually time-corrected recordings.

test type	Prague Labeller	Prak-CV	Prak-HuBERT
phone identity mismatch	6.61%	1.88%	1.12%
match, but misplace ≧ 0.1 s	4.28%	0.36%	0.04%
match, but misplace ≧ 0.2 s	3.22%	0.09%	0.00%
mismatch or misplace ≧ 0.1 s	10.89%	2.24%	1.16%

An improvement can also be observed between the two Prak models, with the HuBERT-based model reducing boundary misplacement errors of 100 ms or more from 0.36% in the Prak-CV model to 0.04%. The new model also virtually eliminates errors involving boundary misplacements of 200 ms or more, which suggests that word-level identification by this model is highly reliable.

Prak-CV and Prak-HuBERT use the same pronunciation module. However, this module often generates variant pronunciations, and Prak-HuBERT makes better use of the acoustic evidence to select the correct variant. This explains the improved performance of Prak-HuBERT, even in terms of phone identification. The phone identity mismatch could be further reduced by using the pronunciation exceptions file. This is an expected practice when using Prak for research purposes, however, we did not generate a dedicated exceptions file for our test set.

Results of additional testing using the Prak-HuBERT model with test data modified to comply with our selected labeling method are shown in Table 3. Compared to the initial test, boundary misplacements of 100 ms or more were reduced ten times, and phone identity mismatch decreased by 0.3 percent following this adjustment. The additional test for boundary misplacements of 50 ms or more can be interpreted as capturing all major shifts, including ones smaller than the typical duration of one phone.

Table 3 Percentage of phone mismatch and boundary misplacement, comparing the output of Prak's HuBERT model with manually time-corrected recordings further double checked for errors and compliance with the selected transcription method.

test type	Prak-HuBERT
phone identity mismatch	0.946%
match, but misplace ≧ 0.05 s	0.133%
match, but misplace ≧ 0.1 s	0.005%
match, but misplace ≧ 0.2 s	0.000%
mismatch or misplace ≧ 0.1 s	0.951%

Overall, the results demonstrate that the HuBERT-based version of Prak significantly improves alignment accuracy over both the earlier Prak-CV model and the Prague Labeller. Reductions in phone identity mismatches and major boundary misplacements indicate a strong alignment with manually annotated data, while additional tests suggest that remaining errors are minimal and approach the limits of what can be reliably measured. These findings support the practical applicability of Prak-HuBERT in precise phone labelling for phonetic research in Czech.

6. Additional remarks

6.1 Public availability

We have decided to make the improved version of Prak, incorporating HuBERT and fine-tuning on additional time-aligned data, freely available for any purpose, with the only added requirement being the citation of relevant publications. The user of Prak now has two options:

1. Use the original model trained on CommonVoice, in which case Prak is available under the very permissive MIT license.
2. Opt for increased precision in phone boundary alignment, closer to practices established by the Institute of Phonetics, Charles University in Prague. In this case, the only additional requirement is that any publication benefiting from the improved model should cite the relevant publications of the Institute of Phonetics, as stated on the license page at time of installation.

Download of the software, along with usage details for both models, is available through the project website on GitHub: https://github.com/vaclavhanzl/prak

6.2 Future work

Prak was designed for the alignment of relatively short recordings, typically around one minute in length. Some internal algorithms (such as the search for the best alignment path and attention evaluation in the transformer layers) have roughly quadratic complexity, which makes processing slow for exceedingly long inputs. Manually splitting recordings into smaller parts is a simple workaround, but automating this step would be a much more user-friendly solution. This will require some additional research, as both the audio and the corresponding text must be divided into corresponding chunks, and the chunks must be large enough to preserve the contextual benefits provided by the transformer-derived embeddings. Nevertheless, such a feature would undoubtedly be welcome by users.

Additional testing of Prak's current and potential future models can also be conducted, for example examining the relevance of phone boundary precision in common phonetic measurements such as formant detection. Outputs based on measurement from purely automatically aligned data could be compared with results obtained using manually corrected time boundaries, providing insight into the level of precision required for reliable acoustic measurements in phonetic research.

Acknowledgments

We would like to thank the Institute of Phonetics, Charles University, for providing the training dataset of manually time-aligned recordings. Without the tedious work of all who have contributed to the corpus, training a model in order to reduce the amount of labor included in phone alignment for the purposes of phonetic research would not be possible.

REFERENCES

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., & Weber, G. (2019). Common Voice: A Massively-multilingual speech corpus. *arXiv Preprint arXiv:1912.06670*.

Bavarian Archive for Speech Signals. (2018). *Terms of Usage*. Version 4.0. BAS Web Services. https://clarin.phonetik.uni-muenchen.de/BASWebServices/help/termsOfUsage

Boersma, P., & Weenink, D. (2023). *Praat: Doing phonetics by computer.* [Computer program]. Version 6.3.14. http://www.praat.org

Boersma, P., Weenink, D., & collaborators. (2023). *Praat: Doing phonetics by computer* [Source code]. https://github.com/praat/praat

Boigne, J. (2021). *HuBERT: How to Apply BERT to Speech, Visually Explained.* https://jonathanbgn.com/2021/10/30/hubert-visually-explained.html

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining*, *10*(1), 35. https://doi.org/10.1186/s13040-017-0155-3

Hanžl, V. (2023). *Details of the Montreal FA*. Prak Wiki. https://github.com/vaclavhanzl/prak/wiki/Details-of-the-Montreal-FA

Hanžl, V., & Hanžlová, A. (2023). Prak: An automatic phonetic alignment tool for Czech. In R. Skarnitzl & J. Volín (eds.), *Proceedings of the 20th International Congress of Phonetic Sciences* (pp. 3121–3125). Guarant International.

Hanžl, V., & Hanžlová, A. (2025). *prak: Czech phonetic alignment tool.* https://github.com/vaclavhanzl/prak

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3451–3460. https://doi.org/10.1109/TASLP.2021.3122291

Kisler, T., Reichel, U., & Schiel, F. (2017). Multilingual processing of speech via web services. *Computer Speech & Language*, *45*, 326–347. https://doi.org/10.1016/j.csl.2017.01.005

Kuldanová, P., Hebal-Jezierska, M., & Petráš, P. (2022). *Orthoepy of West Slavonic Languages (Czech, Slovak and Polish)*. Ostravská univerzita. https://doi.org/10.15452/Ortoepieen.2022

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111–132. https://doi.org/10.1016/j.aiopen.2022.10.001

Machač, P., & Skarnitzl, R. (2009). *Principles of phonetic segmentation*. Epocha.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017). Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. *Proceedings of Interspeech 2017*, 498–502. https://doi.org/10.21437/Interspeech.2017-1386

Opensource.org. (2025). *The MIT License*. Open Source Initiative. https://opensource.org/licenses/MIT

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative style, high-performance deep learning library. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 8026–8037.

Patc, Z., Mizera, P., & Pollak, P. (2015). Phonetic segmentation using KALDI and reduced pronunciation detection in causal Czech speech. *Text, Speech, and Dialogue*, 433–441.

Pavlík, R. (2009). A Typology of assimilations. *SKASE Journal of Theoretical Linguistics*, *6*(1), 2–26.

Pettarin, A. (2018). *A collection of links and notes on forced alignment tools.* https://github.com/pettarin/forced-alignment-tools

Pollák, P., Volín, J., & Skarnitzl, R. (2005). Influence of HMM's parameters on the accuracy of phone segmentation–evaluation baseline. *Proceedings of the 16th Conference Joined with the 15th Czech-German Workshop 'Speech Processing'*, *1*, 302–309.

Pollák, P., Volín, J., & Skarnitzl, R. (2007). HMM-based phonetic segmentation in Praat environment. *The XII International Conference Speech and Computer – SPECOM*, 537–541.

Schiel, F. (1999). Automatic phonetic transcription of non-prompted speech. *Proceedings of the XIVth International Congress of Phonetic Sciences*, 607–610.

Skarnitzl, R. (2011). *Znělostní kontrast nejen v češtině*. Epocha.

Torchaudio Contributors. (2024). *HUBERT_BASE.* https://docs.pytorch.org/audio/2.4.0/generated/torchaudio.pipelines.HUBERT_BASE.html

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Volín, J. (2012). Jak se v Čechách 'rázuje'. *Naše řeč*, *95*(1), 51–54.

Volín, J., & Skarnitzl, R. (2018). *Segmentální plán češtiny*. Univerzita Karlova, Filozofická fakulta.

Volín, J., & Skarnitzl, R. (2022). The impact of prosodic position on post-stress rise in three genres of Czech. *Speech Prosody 2022*, 505–509. https://doi.org/10.21437/SpeechProsody.2022-103

Yang, Y.-Y., Hira, M., Ni, Z., Chourdia, A., Astafurov, A., Chen, C., Yeh, C.-F., Puhrsch, C., Pollack, D., Genzel, D., Greenberg, D., Yang, E. Z., Lian, J., Mahadeokar, J., Hwang, J., Chen, J., Goldsborough, P., Roy, P., Narenthiran, S., … Shi, Y. (2022). TorchAudio: Building blocks for audio and speech processing. *ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6982–6986.

RESUMÉ

Článek představuje nástroj Prak, určený pro automatické časové zarovnání hlásek v češtině, který klade důraz na transparentní modulární strukturu a fonetickou přesnost. Kromě výslovnostního modulu pracujícího s pravidly a seznamem výslovnostních výjimek zavádí Prak nové využití nedeterministických, zpětně postupujících konečných překladových automatů (FST), zejména pro modelování regresivní asimilace v konsonantických shlucích. Dalším inovativním prvkem je integrace modelu HuBERT a trénování na rozsáhlém korpusu manuálně časově zarovnaných nahrávek, čímž se zvyšuje přesnost klasifikace hlásek, aniž by byla ovlivněna náročnost instalace a použití nástroje. Porovnání časového zarovnání hlásek s testovacím korpusem manuálně segmentovaných nahrávek ukázalo, že rozšířený model je výrazně přesnější v porovnání s předchozím Prak-CV modelem i dřívějším dlouhodobě používaným nástrojem pro časové zarovnání hlásek. Nový model výrazně snižuje pravděpodobnost výskytu hrubých chyb v určení hranic i nesouladů v identifikaci hlásek, čímž se úroveň zarovnání přibližuje standardům manuální segmentace. Nástroj je určen nejen fonetikům zabývajícím se češtinou, ale i vývojářům pracujícím s jazyky s podobnou strukturou.

Adléta Hanžlová
Institute of Phonetics
Faculty of Arts, Charles University
Prague, Czech Republic
adleta.hanzlova@ff.cuni.cz

Václav Hanžl
Department of Informatics and Chemistry, Department of Biochemistry and Microbiology
University of Chemistry and Technology Prague
Prague, Czech Republic