Experiments on Kaldi-based Forced Phonetic Alignment for Brazilian Portuguese

Cassio Batista[0000−0001−6799−6058] and Nelson Neto[0000−0003−0408−4187]

Computer Science Graduate Program, FalaBrasil Group, Federal University of Pará
Augusto Corrêa 1, Belém 66075–110, Brazil
{cassiotb,nelsonneto}@ufpa.br

Abstract. Forced phonetic alignment (FPA) is the task of associating a given phonetic unit to a timestamp interval in the speech waveform. Phoneticians are able mark the boundaries with precision, but as the corpus grows it becomes infeasible to do it by hand. For Brazilian Portuguese (BP) in particular, only three tools appear to perform FPA: EasyAlign, Montreal Forced Aligner (MFA), and UFPAlign. Therefore, this work aims to develop resources based on Kaldi toolkit for UFPAlign, including their release alongside all scripts under open licenses; and to bring forth a comparison to the other two aforementioned aligners. Evaluation took place in terms of the phone boundary metric over a dataset of 385 hand-aligned utterances, and results show that Kaldi-based aligners perform better overall, and that UFPAlign models are more accurate than MFA's. Furthermore, complex deep-learning-based approaches did not seem to improve performance compared to simpler models.

Keywords: Forced phonetic alignment · Speech segmentation · Acoustic modeling · Kaldi · Brazilian Portuguese

1 Introduction

The analysis of the prosodic structure of speech very often requires the alignment of the speech recording with a phonetic transcription of the speech, a task known as forced phonetic alignment (FPA). However, transcribing and aligning several hours of speech by hand is very time-consuming, even for experienced phoneticians. As several approaches have been applied to automate this process, some of them brought from the automatic speech recognition (ASR) domain, the combination of hidden Markov models (HMM) and Gaussian mixture models (GMM) has been for long the most widely explored for FPA.

With respect to ASR-based frameworks, we found only three forced aligners that provide pre-trained models for Brazilian Portuguese (BP): EasyAlign [9], Montreal Forced Aligner (MFA) [16] and UFPAlign [32,6]. To the best of our knowledge, EasyAlign is the only HTK-based aligner that ships with a model for BP, MFA is the only Kaldi-based one, and UFPAlign has been evolving through time to work with both HTK and Kaldi as back-end.

It should be remarked that UFPAlign was born in [32] as an early effort to mitigate the gap for Brazilian Portuguese, providing a package with grapheme-to-phoneme (G2P) converter, syllabification system and GMM-based acoustic models trained over the HTK toolkit [34]. As usual, tests comparing the automatic versus manual segmentations were performed. An extra comparison was made to EasyAlign [9], which to our knowledge was the only aligner that supported BP at that moment. It was observed that the tools achieved equivalent behaviors, considering two metrics: boundary-based and overlap rate.

Later on, following Kaldi's success as the *de facto* open-source toolkit for ASR [25] due to its efficient implementation of deep neural networks (DNN) for hybrid HMM-DNN acoustic modeling, UFPAlign was updated in [6] w.r.t. its older HTK-based version, yielding better results with both monophone and triphone GMM-based models, as well as with a standard feed-forward network trained using `nnet2` recipes. Both HTK- and Kaldi-based versions of UFPAlign were then evaluated over a dataset containing 181 utterances spoken by a male speaker, whose phonemes were manually aligned by an expert phonetician.

Therefore, as `nnet2` recipes became outdated, this work builds upon [6] by updating training scripts to Kaldi's `nnet3` recipe, which contains the current state-of-the-art scripts for ASR. Up-to-date versions of the acoustic models, phonetic and syllabic dictionaries were released to the public under the MIT license on FalaBrasil's GitHub account[1], as well as the scripts to generate them. Assuming Kaldi is pre-installed as a dependency, UFPAlign pipeline's works fine under Linux environments via command line, but also provides a graphical interface as a plugin to Praat [3], a popular free software package for speech analysis.

Additionally, some intra- and inter-evaluation procedures were performed, the former considering all acoustic models trained within the Kaldi's default GMM and DNN pipeline, while the latter applied the HTK former version of UFPAlign [32], EasyAlign [9], and MFA [16] aligners over the same dataset for the sake of a fair comparison. The evaluation dataset was extended from 193 utterances spoken by a male individual to include 192 sentences spoken by a female speaker, i.e., 385 manually aligned audio files in total. The similarity measure is given by the absolute difference between the forced alignments with respect to manual ones, which is called phonetic boundary [16].

In summary, the contributions of this work include:

- Release of monophone-, triphone-, and DNN-based (`nnet3`) acoustic models, which comprise a total of five pre-trained, Kaldi-compatible models included as part of UFPAlign. Scripts used to train such models are also available.
- Generation of multi-tier TextGrid files for Praat, based on phonetic and syllabic dictionaries built over a list of words in BP collected from multiple sources and post-processed by GNU Aspell [2] spell checker.
- Comparison to the only two ASR-based phonetic aligners that exist for Brazilian Portuguese (to the best of our knowledge), regarding the phone boundary metric [16] over a dataset of 385 hand-aligned utterances.

[1] https://github.com/falabrasil

The remainder of this paper is as follows. Section 2 presents the FPA procedure with Kaldi, and some other resources used for training and evaluation. Evaluation tests and results are reported and discussed on Sections 3 and 4, respectively. Finally, Section 5 presents the conclusion and plans for future work.

2 Methodology

This section details the forced phonetic alignment process within UFPAlign, which is similar to a traditional decoding stage in speech recognition where one needs an acoustic model and a phonetic dictionary (or lexicon) to decide among senones, except the language model is not necessary in such case.

UFPAlign uses Kaldi as the ASR back-end, and FalaBrasil's grapheme-to-phoneme (G2P) and syllabification tools to provide phonemes and syllables from regular words (also known as graphemes), given that users themselves provide such transcriptions as input alongside with the corresponding audio file. The output is stored in a TextGrid file — a well-known file format for Praat users.

2.1 Kaldi, Grapheme-to-Phoneme and Syllabification Tools

Kaldi [25] is an open-source toolkit developed to support speech recognition researchers. The DNN training framework is provided by Kaldi in three distinct setups[2]: `nnet1` [14], `nnet2` [35,27] and `nnet3`. Unlike `nnet1` and `nnet2`, `nnet3` offers an easier access to more specialized kinds of networks other than simple feed-forward ones, including long short-term memory (LSTM) [21] and time-delay neural networks (TDNN) [22,24], for example.

Scripts in Kaldi's `nnet3` setup use factorized time-delay neural networks (TDNN-F) as default architecture [22], which are a type of feed-forward network that has a behavior similar to recurrent topologies like LSTMs in the sense of capturing past and future temporal contexts w.r.t. the current speech frame to be recognized, but with an easier procedure for parallelization. This opposes to previous `nnet2` recipes, for instance, which are pure vanilla networks.

As Kaldi requires a lexicon to serve as the target being modeled by HMMs, this work uses a G2P converter provided by the FalaBrasil Group as an open-source library written in Java [30,18]. This tool relies on a stress determination system to provide only one pronunciation per word, which means it does not consider co-articulation between words (i.e., cross-word events are ignored). The phonetic alphabet is composed by 38 phonemes plus a silence phone, inspired by the Speech Assessment Methods Phonetic Alphabet (SAMPA) [7].

The syllabification tool, on the other hand, is not a requirement when training acoustic models for ASR, but rather just a feature of UFPAlign for composing another tier in the TextGrid output file. It is also provided by the FalaBrasil Group within the same library as the G2P [19].

[2] http://www.kaldi-asr.org/doc/dnn.html

2.2 Training Speech Corpora and Lexicon

To build an effective acoustic model (AM), a relatively large amount of labeled data is required, apart from a language model (LM) and a pronunciation model (a.k.a. phonetic dictionary or lexicon). An LM is necessary for speech recognition despite not being explicitly used during phonetic alignment itself. The model used here was built in [18] using SRILM [33] toolkit over approximately 1.5 million sentences from the CETENFolha dataset [12]. The FalaBrasil speech corpora, on the other hand, consists of seven datasets in Brazilian Portuguese with a total of approximately 170 hours of transcribed audio, the same as in [6].

Finally, the phonetic dictionary was created via FalaBrasil G2P tool [30,18] based on a list of words collected from multiple sources on the Internet, including University of Minho's Projecto Natura [1], LibreOffice's VERO dictionary [17], NILC's CETENFolha dataset [12], and FrequencyWords repository based on subtitles from OpenSubtitles [8,20]. GNU Aspell [2] is responsible for checking out the spelling and consequently filtering the huge number of words collected, resulting in approximately 200,000 words in the final list.

2.3 Acoustic Models

The deep-learning-based training approach in Kaldi actually uses the GMM training as a pre-processing stage. For this work, AMs were trained by adapting the recipe for Mini-librispeech dataset [23]. For details on the GMM training pipeline, the reader is referred to [6]. The DNN is trained on the top of the last GMM model of the pipeline, which comprises a speaker-adapted training (SAT).

Figure 1 details how the DNN model is obtained as a final-stage AM by using the neural network to model the state likelihood distributions as well as to

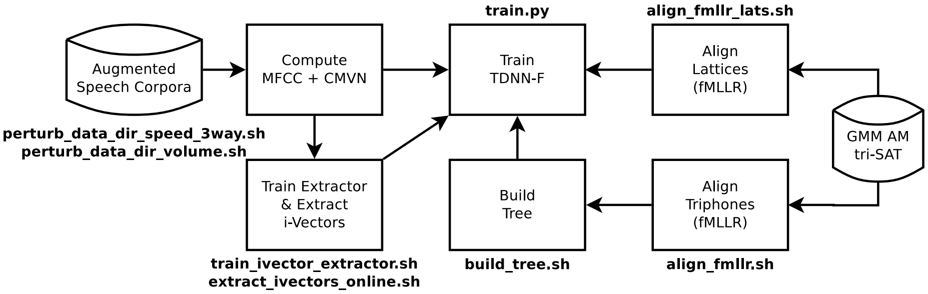

Fig. 1: Stages for training a TDNN-F following Kaldi's Mini-librispeech recipe. On the left side, high-resolution, cepstral-normalized MFCCs (40 features instead of 13) are extracted from an augmented corpora after applying speed and volume perturbation [15], as are the speaker-related 100-dimensional i-vector features [5,31]; to be used as input to the neural network. On the right side, training labels are provided by a GMM tri-SAT acoustic model.

input those likelihoods into the decision tree leaf nodes [10]. The implementation in Kaldi uses a sub-sampling technique that avoids the whole computation of a feed-forward's hidden activations at all time steps and therefore allows a faster training of TDNNs. The "factorized" term distinguishes a TDNN-F from a traditional TDNN architecture by a singular value decomposition (SVD) that is applied at the hidden layer's weight matrices in order to reduce the number of model parameters without degrading performance [24].

2.4 Kaldi Forced Phonetic Alignment

Kaldi's FPA procedure performs several steps for obtaining the time-marked conversation (CTM) files, which contains a list of numerical indices corresponding to phonemes with both their start times and durations in seconds. After Kaldi scripts extract some features from time-domain audio data, the forced alignment step, that employs the aforementioned pre-trained acoustic models, is computed by Kaldi using Viterbi beam search algorithm [11]. Figure 2 shows an overview of the stages within UFPAlign.

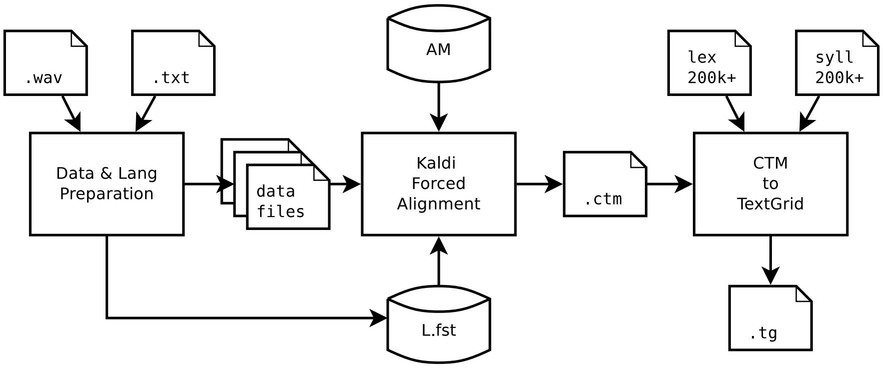

Fig. 2: Pipeline followed by UFPAlign. When a user feeds the system with an audio (`.wav`) and its respective transcription (`.txt`), they should expect a Praat's TextGrid file (`.tg`) as output. Time marks are provided by Kaldi, which relies on the knowledge of the acoustic model (AM) and tokens of the lexicon (`L.fst`).

The data and language preparation stage in particular also creates some "data files" on the fly, which contain information regarding the specifics of the audio file and its transcription, namely `text`, `wav.scp`, `utt2spk`, and `spk2utt`. The language preparation stage, on the other hand, is given by a script provided by Kaldi to create another set of important files, the main one being the lexicon parsed into a finite-state transducer (FST) format, called `L.fst`.

For data preparation, the first step consists in checking whether there are any new words in the input data that were not seen during the acoustic model training. If any word in the transcriptions is not found in the pronunciation dictionary (lexicon), it calls the grapheme-phoneme conversion module (G2P) [30,18] to extend the lexicon with each new word along with its respective phonemic pronunciation. For Praat's final visualization purposes, the word is also divided into syllables through the embedded syllabification tool [19]. As the original phonetic and syllabic dictionaries contain approximately 200,000 entries, they both become `lex 200k+` and `syll 200k+` files after the insertion of missing words.

The last block of the phonetic alignment process handles the conversion of both CTM files to a Praat's TextGrid (`.tg`), a text file containing the alignment information. Therefore, CTM files are read by a Python script that in the conversion process uses the `lex 200k+` and `syll 200k+` extended dictionaries to generate the output five-tier TextGrid that can be displayed by Praat's editor.

3 Evaluation Tests

The evaluation procedure takes place by comparing a bunch of TextGrid files: the hand-aligned reference and the ones automatically annotated by the forced aligners (i.e., by inference), as the phone boundary metric considers the absolute difference between the ending time of both phoneme occurrences [16]. The calculation is performed for each acoustic model, and it takes place over all utterances from the evaluation dataset composed by one male and one female speaker.

3.1 Evaluation Speech Corpus

The automatic alignment was estimated on the basis of the manual segmentation. The original dataset used for assessing the accuracy of the phonetic aligner is composed of 200 and 199 utterances spoken by a male and a female speaker, in a total of 15 minutes and 32 seconds of hand-aligned audio, as shown in Table 1. Praat's TextGrid files, whose phonetic timestamps were manually adjusted by a phonetician, are available alongside audio and text transcriptions.

Table 1: Speech corpus used to evaluate the automatic phonetic aligners. Actual duration and number of files after discard are shown between parentheses, as well as the number of unique words.

Dataset	Duration	# Files	# Words	# Tokens
Male	7m:58s (7m:40s)	200 (193)	1,260 (665)	5,275
Female	7m:34s (7m:18s)	199 (192)	1,258 (664)	5,262
Total	15m:32s (14m:58s)	399 (385)	2,518 (686)	10,537

This dataset was aligned with a set of phonemes inspired by the SAMPA alphabet, which in theory is the same set used by the FalaBrasil's G2P software that creates the lexicon during acoustic model training. Nevertheless, there are some problems of phonetic mismatches, and some cross-word phonemes between words, which makes the mapping between both phoneme sets challenging, given that FalaBrasil's G2P only handles internal-word conversion [30].

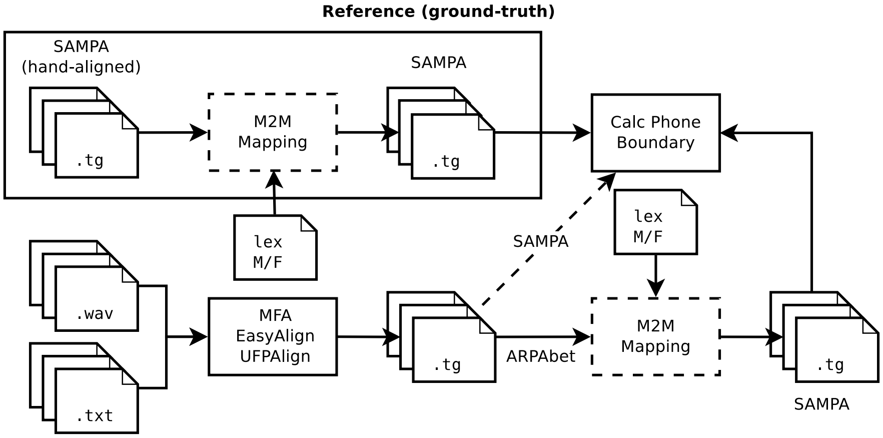

Fig. 3: Evaluation takes place by comparing the output of all forced aligners to a hand-aligned ground-truth. The M2M mapping is applied to make different phone sets match the SAMPA version used by FalaBrasil's G2P, which is provided by the lexicon generated over transcriptions of the corpus (`lex M/F`).

The example in Table 2 shows the phonetic transcription for a sentence given by the original dataset (top) and the acoustic model (bottom) which then suppress vowel sounds altogether due to cross-word rules (usually elision and apocope) when they occur at the end of the current word and at the beginning at the next. Such mismatches occur because the dataset was aligned by a phonetician considering acoustic information (i.e., listening), which cannot be done by the G2P tool that creates the acoustic model's lexicon, since it is provided only with textual information. Situations like these of phonetic information loss led to the removal of such audio files from the dataset before evaluation.

Table 2: Cross-word mismatches between transcriptions manually aligned by a phonetician (top) vs. generated by our G2P software (bottom). Word boundary losses are represented by the empty set symbol (∅).

(a) “***às novi*** *meia,* ***pairum ar*** *no rio*” → “***às nove e*** *meia,* ***paira um ar*** *no rio*”

6	∅	Z	n	O	v	i	∅	...	p	a	j	4	∅	u~	m	a	h/ ...
a	j	s	n	O	v	i	i	...	p	a	j	r	a	u~	∅	a	X ...

In the end, fourteen files were excluded from the dataset, so about 34 seconds of audio was discarded, and 193 and 192 utterances remained in the male and female datasets, respectively. The filtering also ignored intra- and inter-word pauses and silences, resulting in 2,518 words (686 unique, since the utterances' transcriptions are identical for both speakers, i.e, they speak the very same sentences) and 10,537 phonetic segments (tokens) (c.f. Table 1).

3.2 Simulation Overview

Figure 3 shows a diagram of the experiments where EasyAlign, UFPAlign and MFA forced aligners receive the same input of audio files (.wav) with their respective textual transcriptions (.txt). These are the files whose manual annotation is available. All three aligners output one TextGrid file (.tg) for each audio given as input, which then serve as the inference inputs to the phone boundary calculation. The reference ground-truth annotations, on the other hand, are provided by the 385 TextGrid files that contain the hand-aligned phonemes corresponding to the transcriptions in the evaluation dataset.

However, for computing phone boundaries, there must exist a one-to-one mapping between the reference and the inference phones, which was not possible at first due to the nature of the phonetic alphabets: UFPAlign and EasyAlign share the same SAMPA-inspired lexicon generated by FalaBrasil's G2P tool, while MFA is based on ARPAbet [29]. Furthermore, the hand-aligned utterances fall on a special case where the phonetic alphabet used (referred here as "original") is also SAMPA-inspired, but not exactly the same as FalaBrasil's.

Apart from the fact that cross-word rules can insert or delete phones, some phonemes do not have an equivalent, such as /tS/ and /dZ/. Besides, there are also usual swaps between phonetically similar sounds: /h//, /h\/, /h/ and /4/, for instance, might be almost deliberately mapped to either /r/, /R/ or /X/. This is worse in MFA, where the set of phonemes is entirely different.

Thus, since the situation seemed to require a smarter approach than a simple one-to-one tabular, static mapping, it was necessary to employ a many-to-many (M2M) mapping procedure (c.f. dashed blocks on Fig. 3) based on statistical frequency of occurrence, e.g., how many times phones /t/ and /S/ from the original evaluation dataset were mapped to a single phone /tS/ in the lex M/F file representing FalaBrasil's G2P SAMPA-inspired alphabet. This mapping also works when dealing with MFA's ARPAbet phonemes, and will be further discussed in Section 3.3.

3.3 Many-to-Many (M2M) Phonetic Mapping

By taking another look at Table 2, one might have also reasoned that the mapping between the two sets of phonemes is not always one-to-one. The usual situation is where a pair of phonemes from the dataset (original) is merged into a single one for the AM (FalaBrasil G2P), such as /i∼/ /n/ → /i∼/ and /t/ /S/ → /tS/. However, a single phoneme can also be less frequently split into two or more, such as /u/ /S/ → /u/ /j/ /s/.

To deal with these irregularities, we used the many-to-many alignment (m2m-aligner) software [13] in the core of a pipeline that converts the original TextGrid from the evaluation dataset to a TextGrid that is compatible with the FalaBrasil's lexicon used to train the acoustic models. We took advantage of the same pipeline to convert MFA's ARPAbet-based phonemes to SAMPA as well.

The m2m-aligner works in an unsupervised fashion, using an edit-distance-based algorithm to align two strings from a file in the news format, in order for them to share the same length [13]. All 385 TextGrid files from our evaluation dataset (`.tg`) are used to compose a single `news` file, as exemplified in Table 3. Notice the file is composed by the phonemes of the whole sentence rather than by isolated words, in order to mitigate the effects of the cross-word boundaries. The string mapping is finished after a certain number of iterations when the m2m-aligner provides a one-to-one mapping in a file we called `m2m` that joins some phonemes together, as shown by shades of gray in Table 3.

Table 3: Example of a single `news` file with phonemes from three out of 385 TextGrid files for sentences *"é bom pousar"* and *"os lindos jardins"*. Each line contains a whole phonetic sentence to be converted, and different phone sets are separated into two distinct columns divided by a tabular `'\t'` character, so every other token is separated by a single space. Groups of phonemes which are supposed to be later merged by m2m-aligner in the `m2m` file are shaded in gray.

(a) Original dataset phone set (original SAMPA) vs. FalaBrasil's (SAMPA)

Dataset phonemes (SAMPA, original)	AM phonemes (SAMPA, FB)
`E b o~ n p o w z a h`	`E b o~ p o w z a X`
`u S l i~ n d u S Z a h\ d Z i~ n S`	`u j s l i~ d u s Z a R dZ i~ s`

(b) MFA phone set (ARPAbet) vs. FalaBrasil's (SAMPA)

MFA phonemes (ARPAbet)	AM phonemes (SAMPA, FB)
`E+ B O~+ W~ P O Z A+ RR`	`E b o~ p o w z a X`
`UX S L I~+ D UX S Z A RR DJ I~ S`	`u j s l i~ d u s Z a R dZ i~ s`

Finally, as the m2m-aligner provides the mapping for phonemes, another script provides the time stamps calculations prior to creating the converted TextGrid file. Table 4 illustrates how the phonetic time stamps, in milliseconds, are mapped accordingly. Basically if two or more phonemes are mapped into a single one (merging), as in `/o~/ /n/` → `/o~/` or `/d/ /Z/` → `/dZ/` (marked with an $*$), the time stamp of the last phoneme is considered. However, if one phoneme is mapped to two or more (splitting) as in `/e~/` → `/e~/ /j~/`, then linearly spaced time stamps are generated in between the phone to be split (†) and its immediate predecessor (‡).

4 Results and Discussion

Results will be reported in terms of statistics such as mean (μ), median and standard deviation (σ) over the distribution of phone boundary values, and a tolerance threshold that shows how many phonetic tokens were more precisely

Table 4: Conversion of time stamps for the sentence "*onde existem*".

494	533*	558	565*	583	682	748	854	929	979‡	1042†	
o~	n	d	Z	i	i	z	i	S	t	e~	
o~		dZ		i	e	z	i	s	t	e~	j~
533*		565*		583	682	748	854	929	979‡	1010	1042†

aligned with respect to the manual alignments. Numerical values, in milliseconds, are presented in Table 5. The best ones are highlighted in bold.

As far as MFA train-and-align (T&A) feature is concerned, roughly only 1% of phoneme tokens aligned by Kaldi-based aligners are off the 100 ms tolerance, against 3% of tokens aligned by HTK-based tools. In fact, approximately 96%–97% of phonemes were under the 50 ms tolerance when aligned by acoustic models trained with MFA and UFPAlign, considering an average of all models. Unfortunately, this is not true for MFA's pre-trained model for Brazilian Portuguese (in align-only mode), which on the other hand, for larger tolerance threshold values, performed a little worse than HTK.

Among HTK-based aligners, EasyAlign performed best considering all statistics and tolerance thresholds for both male and female speakers. However, as already pointed out in [32], the same ground-truth dataset used for evaluation in this work was also used to train the BP acoustic model shipped with EasyAlign, so this might have had some bias during the comparison. Overall, UFPAlign (HTK) achieved very similar values across metrics for both speakers of the dataset, while EasyAlign's behavior shows a greater accuracy on the female voice. Nevertheless, the parcel of phonetic tokens whose difference to the manual segmentation was less than 10 ms stayed below the 40% even for EasyAlign.

In align-only (A) mode, MFA models performed slightly better until 10 ms than EasyAlign's, but increasingly worse for larger values of tolerance for both male and female speakers. These poor results may be due to the nature of the dataset used to generate MFA's pre-trained acoustic models (GlobalPhone [28]), which contains only 22 hours of transcribed audio. In contrast, training and aligning (T&A) on the same evaluation dataset with MFA proved better than HTK for the male speaker, and the results are similar for the female speaker.

The monophone- and triphone-based GMM models we trained with Kaldi for UFPAlign achieved the best performance with respect to phone boundary when compared to both MFA and HTK-based aligners. On average, approximately 45% of tokens were accurately aligned within the 10 ms margin for all GMM models. Mean and median values are the lowest (except for tri-SAT on the male dataset, which was greater than MFA's T&A) and at most ∼4 ms distant from each other. With respect to the speakers' gender, UFPAlign (Kaldi) performed approximately 4% better for the woman's voice until the 50 ms of tolerance, and about 2 ms more accurate according to the average mean.

Finally, TDNN-F simulation was definitely disappointing. We expected that results from a `nnet3` DNN-based setup would be at least similar to GMM-based

Table 5: Results regarding mean (μ), median (med.), standard deviation (σ), and cumulative percentage below a tolerance threshold, in milliseconds, of the differences between forced aligned audio and ground-truth (hand aligned) phonemes, also known as phone boundary. Notations on MFA stand for align-only (A) and train-and-align (T&A) procedures, while on UFPAlign they denote either the nature of the toolkit or the acoustic model.

					Cumulative tolerance (%)			
	Toolkit	μ	med.	σ	<10 ms	<25 ms	<50 ms	<100 ms
Female dataset	UFPAlign (HTK)	26.44	17.00	38.31	31.40	63.94	88.19	97.08
	EasyAlign	18.42	13.00	20.30	36.59	78.12	94.06	98.91
	MFA (A)	23.62	12.00	34.16	39.34	75.99	87.77	95.65
	MFA (T&A)	17.60	13.00	18.62	37.65	78.69	95.16	99.08
	UFPAlign (mono)	13.58	10.00	15.02	47.47	87.70	97.55	99.57
	UFPAlign (tri-Δ)	12.43	9.00	13.28	**50.44**	**89.88**	**98.34**	99.62
	UFPAlign (tri-LDA)	12.99	10.00	12.62	47.48	89.22	98.27	99.76
	UFPAlign (tri-SAT)	13.43	10.00	12.75	45.69	88.20	98.15	99.77
	UFPAlign (TDNN-F)	17.18	14.00	13.87	34.41	75.94	97.61	**99.87**
Male dataset	UFPAlign (HTK)	26.86	17.00	32.61	30.73	62.45	86.55	96.42
	EasyAlign	24.35	17.00	30.70	31.53	67.51	89.69	96.95
	MFA (A)	34.28	16.00	46.70	32.81	64.85	78.49	90.61
	MFA (T&A)	14.65	11.00	14.37	45.12	83.34	**97.23**	**99.66**
	UFPAlign (mono)	15.25	11.00	15.70	43.51	83.42	96.29	99.42
	UFPAlign (tri-Δ)	14.16	10.00	14.06	**46.28**	**85.55**	97.13	99.74
	UFPAlign (tri-LDA)	14.66	11.00	13.82	43.49	84.50	97.19	99.74
	UFPAlign (tri-SAT)	14.96	12.00	13.77	42.14	83.51	97.19	99.78
	UFPAlign (TDNN-F)	18.58	16.00	14.26	32.02	70.62	96.65	99.94

ones, as it was in [6] with `nnet2`, but cumulative tolerance values were instead just slightly better than EasyAlign. Therefore, even though one can say that the best result was achieved by tri-delta (Δ) models on both male and female datasets, since it holds the rows with most boldface values in Table 5 (except MFA was better off after 50 ms on the man's voice, but the values compared to UFPAlign's tri-Δ model are fairly and virtually the same), we would rather prefer to state that all GMM-based AMs in UFPAlign achieved similar results. Even monophone models, the simplest ones, had a close performance on tri-SAT, the most complex.

4.1 Discussion

A possible reason for such a difference between HTK- and Kaldi-based aligners might be that HTK uses Baum-Welch algorithm for training HMMs while Kaldi uses Viterbi training [4]. On the other hand, among Kaldi models, tri-Δ stands out as being virtually the best one. However, with just a $\sim$1–3% difference in tolerance, and $\sim$1 ms difference in both mean and median values, we cannot tell whether it is significant enough to classify one model into being better than the others, as they appear pretty close at glance. The linear sequence of model training just does not result in lower errors in phonetic boundaries as it resulted in lower word error rates for speech recognition.

The somewhat shocking results were produced by the DNN. For the state of the art for ASR to perform so poorly in phonetic alignment problems, it certainly needs careful investigation. We suspect the HMM topology used in `nnet3` chain models, which can be traversed in one frame rather than in three on the traditional left-to-right [26], may have had some unfavorable influence. Moreover, data insufficiency could even have been the problem for the DNN in the first place, since the $\sim$171 hours in our training dataset are far from the ideal volume to train a neural network efficiently. Other reasons include the possible high number of hidden layers in the TDNN-F, and the use of frame subsampling, which requires an extra normalization value to be passed to Kaldi's `ali-to-phones` script for compensation.

Besides, navigating through all the burden to train a DNN model with Kaldi (which requires at least one GPU card) may not be the more appropriate move if the final task's goal is to align phonemes rather than to recognize speech. As MFA seem to have dropped support to DNN models, and our previous results with a `nnet2` neural network setup only took tolerance values so far as to match tri-Δ models [6], we feel discouraged to invest so much time computer power to train a DNN model. Nevertheless, conjectures still need to be experimented.

5 Conclusion

This paper presented contributions for the problem of forced phonetic alignment (FPA) in Brazilian Portuguese (BP). An update to UFPAlign [6] was offered by providing adapted Kaldi recipes for training acoustic models on BP datasets, as well as properly releasing all the acoustic models for free under an open-source license on the GitHub of the FalaBrasil Group[3]. UFPAlign works either via command line (Linux) or in a graphical interface as a plugin to Praat. Up-to-date phonetic and syllabic dictionaries created over a list of 200,000 words for BP are also provided, as well as standalone grapheme-to-phoneme and syllabification systems for handling out-of-vocabulary words.

For evaluation, a comparison among the Kaldi-based acoustic models trained with an updated version of the scripts from [6] was performed, as well as a comparison to an outdated HTK-based version of UFPAlign from [32]. Results

[3] https://github.com/falabrasil

regarding the absolute difference between forced and manual aligned utterances (phone boundary metric) showed that the HTK-based aligner performed worse when compared to any of the Kaldi-based models, and that our acoustic models we trained from scratch performed better than MFA's pre-trained models.

5.1 Future Work

As future work, there are a couple of experiments to be investigated. The simplest one would be to train GMM-based tri-Δ, tri-LDA, tri-SAT and even monophone-based acoustic models with a higher number of Gaussian mixtures per senone. Training a DNN on the top of tri-Δ, since that was the one that yielded the most accurate results according to phone boundary, should be also worth trying. Besides, training a DNN on the top of context-independent monophones does not sound so absurd either, given the proximity of the results.

Regarding the DNN, one thing to verify is whether removing the i-vectors and leaving just normalized MFCCs as input features would result in more accurate alignments. Splicing cepstral features with LDA would also be a valid test. By the way, the TDNN-F setup has not been altered from Mini-librispeech's default recipe, which means some parameters such as layer dimension, number of layers, context width, and the application of frame subsampling could still undergo tuning. Finally, other architectures like LSTMs should have its use evaluated.

Another idea might be the employment of transfer learning techniques to take advantage of models pre-trained on larger volumes of audio data and just make some adaptations to make it work on our evaluation dataset. That way, an acoustic model trained over LibriSpeech dataset, for example, could be downloaded from OpenSLR [23] to serve as a starting point, and GMM-based models would be trained from scratch over the male/female evaluation dataset to play the role of the new tri-SAT reference alignments. One impediment, however, is that most of the pre-trained TDNN-F-based models available on the Internet are chain models (i.e., a simplified HMM topology is used to model phonemes), which suggests a new, chain-free model would have to be trained from scratch on English data, which is also freely available.

At last, although UFPAlign can be used as a plugin to Praat, we plan in the future to train models compatible with MFA under the same licensing, as to avoid open-source competition. The provision of a train-and-align feature for UFPAlign is also an ongoing plan.

Acknowledgment

We gratefully acknowledge NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors also would like to thank CAPES for providing scholarships and FAPESPA (grant 001/2020, process 2019/583359) for the financial support.

References

1. Almeida, J.J., Simões, A.: Projecto natura (2021), https://natura.di.uminho.pt/wiki/doku.php
2. Atkinson, K.: Gnu aspell (2021), https://aspell.net
3. Boersma, P., Weenink, D.: Praat: doing phonetics by computer (version 6.1.15) [computer program] (2020), available at https://www.fon.hum.uva.nl/praat/
4. Buthpitiya, S., Lane, I., Chong, J.: A parallel implementation of viterbi training for acoustic models using graphics processing units. In: 2012 Innovative Parallel Computing (InPar). pp. 1–10 (2012). https://doi.org/10.1109/InPar.2012.6339590
5. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing **19**(4), 788–798 (2011). https://doi.org/10.1109/TASL.2010.2064307
6. Dias, A.L., Batista, C., Santana, D., Neto, N.: Towards a free, forced phonetic aligner for brazilian portuguese using kaldi tools. In: Cerri, R., Prati, R.C. (eds.) Intelligent Systems. pp. 621–635. Springer International Publishing, Cham (2020)
7. Gibbon, D., Moore, R., Winski, R.: Sampa computer readable phonetic alphabet (2021), available at https://www.phon.ucl.ac.uk/home/sampa/
8. GitHub: Frequencywords (2020), https://github.com/hermitdave/FrequencyWords
9. Goldman, J.P.: Easyalign: An automatic phonetic alignment tool under praat. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. pp. 3233–3236 (01 2011)
10. Guiroy, S., de Cordoba, R., Villegas, A.: Application of the Kaldi toolkit for continuous speech recognition using Hidden-Markov Models and Deep Neural Networks. In: IberSPEECH'2016 On-line proceedings. pp. 187–196. IberSPEECH 2016, Lisboa, Portugal (November 2016)
11. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edn. (2001)
12. Interinstitutional Center for Computational Linguistics: Cetenfolha dataset (2021), https://www.linguateca.pt/cetenfolha/index_info.html
13. Jiampojamarn, S., Kondrak, G., Sherif, T.: Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference. pp. 372–379. Association for Computational Linguistics, Rochester, New York (April 2007), http://www.aclweb.org/anthology/N/N07/N07-1047
14. K, V., A, G., L, B., D, P.: Sequence-discriminative training of deep neural networks. In: INTERSPEECH 2013. pp. 2345–2349 (2013)
15. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: Proceedings of Interspeech (07 2015)
16. McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., Sonderegger, M.: Montreal forced aligner: Trainable text-speech alignment using kaldi. In: Proceedings of Interspeech. pp. 498–502 (08 2017). https://doi.org/10.21437/Interspeech.2017-1386
17. Moura, R.: Libreoffice's vero dictionary (2021), https://github.com/LibreOffice/dictionaries/tree/master/pt_BR
18. Neto, N., Patrick, C., Klautau, A., Trancoso, I.: Free tools and resources for brazilian portuguese speech recognition. Journal of the Brazilian Computer Society **17**(1), 53–68 (Mar 2011). https://doi.org/10.1007/s13173-010-0023-1

19. Neto, N., Rocha, W., Sousa, G.: An open-source rule-based syllabification tool for brazilian portuguese. Journal of the Brazilian Computer Society **21**(1) (2015). https://doi.org/10.1186/s13173-014-0021-9
20. Opensubtitles.org: Opensubtitles (2021), https://www.opensubtitles.org/
21. Peddinti, V., Wang, Y., Povey, D., Khudanpur, S.: Low latency acoustic modeling using temporal convolution and lstms. IEEE Signal Processing Letters **25**(3), 373–377 (2018). https://doi.org/10.1109/LSP.2017.2723507
22. Peddinti, V., Povey, D., Khudanpur, S.: A time delay neural network architecture for efficient modeling of long temporal contexts. In: Proceedings of Interspeech. pp. 3214–3218 (08 2015)
23. Povey, D.: Openslr: Open speech and language resources (2021), https://openslr.org/index.html
24. Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., Khudanpur, S.: Semi-orthogonal low-rank matrix factorization for deep neural networks. In: Proc. Interspeech 2018. pp. 3743–3747 (2018). https://doi.org/10.21437/Interspeech.2018-1417
25. Povey, D., Ghoshal, A., Boulianne, G., Goel, N., Hannemann, M., Qian, Y., Schwarz, P., Stemmer, G.: The kaldi speech recognition toolkit. In: In IEEE 2011 workshop (2011)
26. Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., Khudanpur, S.: Purely sequence-trained neural networks for asr based on lattice-free mmi. In: Interspeech 2016. pp. 2751–2755 (2016). https://doi.org/10.21437/Interspeech.2016-595, http://dx.doi.org/10.21437/Interspeech.2016-595
27. Povey, D., Zhang, X., Khudanpur, S.: Parallel training of dnns with natural gradient and parameter averaging (2015)
28. Schultz, T., Vu, N.T., Schlippe, T.: Globalphone: A multilingual text speech database in 20 languages. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 8126–8130 (May 2013). https://doi.org/10.1109/ICASSP.2013.6639248
29. Shoup, J.E.: Phonological aspects of speech recognition. Trends in speech recognition pp. 125–138 (1980)
30. Siravenha, A., Neto, N., Macedo, V., Klautau, A.: Uso de regras fonológicas com determinação de vogal tônica para conversão grafema-fone em português brasileiro (01 2008)
31. Snyder, D., Garcia-Romero, D., Povey, D., Khudanpur, S.: Deep neural network embeddings for text-independent speaker verification. In: Proc. Interspeech 2017. pp. 999–1003 (2017). https://doi.org/10.21437/Interspeech.2017-620
32. Souza, G., Neto, N.: An automatic phonetic aligner for brazilian portuguese with a praat interface. In: Silva, J., Ribeiro, R., Quaresma, P., Adami, A., Branco, A. (eds.) Computational Processing of the Portuguese Language. pp. 374–384. Springer International Publishing, Cham (2016)
33. Stolcke, A.: Srilm – an extensivle language modeling toolkit. In: Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP). vol. 2, pp. 901–904 (2002)
34. Young, S., Ollason, D., Valtchev, V., Woodland, P.: The HTK Book. Cambridge University Engineering Department, version 3.4, Cambridge, UK (2006)
35. Zhang, X., Trmal, J., Povey, D., Khudanpur, S.: Improving deep neural network acoustic models using generalized maxout networks. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 215–219 (2014). https://doi.org/10.1109/ICASSP.2014.6853589