Alinha-PB: A Phonetic Aligner for Brazilian Portuguese

João G. S. Kruse and Plínio A. Barbosa

Abstract— **Phonetic alignment is the task of finding the limits of phones and higher units in an audio file. This has been reliably done in many languages such as English, French and German, but, so far, no available Brazilian Portuguese aligner had a performance comparable with the ones used for these other languages. Thus, the main goal of this work was to implement a useful tool for forced alignment for Brazilian Portuguese. The implementation was done in two steps, the grapheme-to-phoneme conversion and the alignment itself. The Converter is responsible for receiving the input transcription in graphemes and converting it to its equivalent in phonemes and allophones, and was implemented using computational rules derived from the analysis of regular grapheme-phoneme relations in Brazilian Portuguese and an exception dictionary, for words to which no regular rules could be applied. The Aligner was responsible for aligning the phonemes/allophones of the previous module to the corresponding acoustic intervals of the audio file, called "phones". This module was implemented using hidden Markov models. Results for the Converter have an accuracy of over 99%, where the main mistakes involved mid vowels /e/ and /ɛ/ and /o/ and /ɔ/. As for the Aligner, the best model has 87% of the alignments with errors below 25 ms.**

Index Terms— **Phonetic Annotation, Machine Learning, Brazilian Portuguese.**

I. Introduction

Forced alignment is a category of alignment in which the algorithm receives as an input an audio file and a text containing the orthographic transcription of the audio, and generates as an output the times and the phonetic symbols aligned with the audio and corresponding to each part of the text that was spoken. Forced alignment has many applications for phonetic and speech corpora studies and can also be used to generate datasets for the training of speech recognition algorithms, for prosodic and segmental studies in Experimental Phonetics, Speech-based Corpus Linguistics and Laboratory Phonology, to cite a few applications.

Forced aligners are toolkits that can be trained for any language and indeed some have been created and made available to the public for many languages, such as the Montreal Forced Aligner [18] and webMAUS [21], both with models for several languages, including (European) Portuguese in the case of the former. Also notable is ProsodyLab [14], which is completely accessible via the GitHub platform. Even if these three toolkits allow researchers to develop aligners for any language, there is no freely available forced aligner for Brazilian Portuguese with state-of-the-art acceptable results, so this is the main goal of this work.

A forced aligner can be divided in two core modules, the grapheme-phoneme converter and the aligner itself. The first module is responsible for pre-processing the orthographic text at the input in order to obtain a sequence of phoneme or allophones at the output so that this representation can be used in the second stage as the input to the aligner, which will create the alignment between the phoneme/allophone realization (phone) in a particular audio interval attributing the appropriate labels delivered by the first module.

To implement the aligner module, Hidden Markov Models [12] are the most used and accurate method for recognizing phonetic sequences with the publicly available HTK [22] and Kaldi toolkits [20] for implementation. These two toolkits were used in this work to create the aligners presented here.

The results of an HMM tend to be more speaker-dependent, so that the created model performs better for speakers with phonetic similarities to the speakers used for training the model. Although deep neural networks [15] tend to work better in terms of generalization, it needs huge datasets, which is not our case. An advantage of building a forced aligner in the state-of-the-art with modest dataset sizes, provided they cover sufficient representativity of phonetic variation, is the gain in terms of hours dedicated to research. Thus, HMM was our choice for implementing Alinha-PB, the phonetic aligner for Brazilian Portuguese.

In the following, the two components of Alinha-PB are described: the converter and the aligner, including their validation and a first set of tests carried out on both the training set and the test set. The final sections present the results, a general discussion and give the details about the availability of the tool.

II. Converter

The Alinha-PB grapheme-to-phoneme converter converts a text written orthographically to its corresponding phoneme/allophone representation for Brazilian Portuguese, and includes the possibility of grouping these segments into words and phonetic syllables. To do so, and following

This work was supported in part by the CNPq under Grant n. 302194/2019-3 to the second author and a PIBIC grant to the first author.

João G. S. Kruse is with the Institute of Computing, University of Campinas, Brazil (e-mail: j218615@dac.unicamp.br).

Plínio A. Barbosa is with the Department of Linguistics, University of Campinas, Brazil (e-mail: pabarbosa.unicampbr@gmail.com).

Digital Object Identifier: 10.14209/jcis.2021.21

mainstream literature on knowledge-based converters (see Hunnicutt [16] for a review), we used computational production rules of the form A -> B/L_R (grapheme A rewrites to phoneme or allophone B if A is in the left context L and right context R) and an exceptions dictionary for the words for which the production rules do not apply. For the sake of portability and use in platforms such as Praat [8], the phonetic symbols generated at the output level are coded in ASCII, which presents the respective IPA correspondence given in Tab. I.

TABLE I: Correspondence between IPA symbols and symbols used in Alinha-PB

IPA	ASCII	IPA	ASCII	IPA	ASCII
i	i	ej	eI	p	p
e	e	ɛj	ehI	t	t
ɛ	eh	aj	aI	k	k
a	a	ɔj	ohI	b	b
ɔ	oh	oj	oI	d	d
o	o	uj	uI	g	g
u	u	ɐ̃j̃	aNI	f	f
ĩ	iN	õj̃	oNI	s	s
'ẽ	eN	iw	iU	ʃ	sh
'ɐ̃	aN	ew	eU	v	v
'õ	oN	ɛw	ehU	z	z
'ũ	uN	aw	aU	ʒ	zh
ɪ	I	ɔw	ohU	s de coda	S
e*	E	ow	oU	m	m
ɐ	A	'ɐ̃w̃	aNU	n	n
o**	O	ɪw	IU	ɲ	nh
ʊ	U	ʊw	UU	/r/	r
ĩ	IN	ɪj	II	R (ɾ, ɹ, ɽ)	R
ẽ	EN	ʊj	UI	l	l
ɐ̃	AN	ɪɐ	IA	ʎ	lh
õ	ON	wɐ	UA	ɫ	L
ũ	UN	ɐ̃w̃	ANU		

* as in "ópera"
** as in "cômodo"

A. *Rules for conversion*

The rules were created based on grapheme-phoneme correspondences found in Albano and Moreira [1] supplemented by a search in corpora, especially the Corpus Brasileiro [4] for specific pronunciation rules that would deal with the majority of high-frequency words in the case of the mid vowels /e/ vs. /ɛ/ (orelha vs. velha) and /o/ vs. /ɔ/ (poço vs. posso). The choice for a particular rule was guided for the sake of having a large coverage of word tokens with a particular pronunciation. To determine the rules, several searches in free dictionary entries in the Internet (such as www.dicio.com.br) for large sets of words containing the grapheme sequences to be converted were analysed and a rule was created so that it correctly predicted the phoneme representation for the largest number of the words. During this process, we realized that certain words with very similar graphemic sequences have different conversions, which brought about the need for an exceptions dictionary. For the application of the rules themselves, it is first necessary to find the position of stress in the word.

B. *Pre-processing: finding stress position*

To determine where stress is in the word, a process similar to the complementary use of rules for graphic accentuation in Portuguese was applied, as follows. The first step is to verify if the word has one of the following accentuation marks the circumflex or the acute diacritics and, if so, this syllable was considered stressed. If not, we verify if the word has a tilde mark, and, if so, that syllable is stressed, because it is heavy [7]. If none of these conditions are found, we know that the word is either a paroxytone (stress on the second last syllable) or an oxytone (stress on the last syllable), so we can restrict our analysis to them. If a word is oxytone, we know that it is graphically accentuated when it ends in “a”, “as”, “o”, “os”, “e”, “es”, “em”, “ens”, and because of that, if the word ends in one of those sequences and is not marked with an accent diacritic, we know it must be a paroxytone. By the same rule, if a word does not end in one of those sequences and is not graphically accentuated, it must be an oxytone (the only exceptions to this rule, which can be easily managed, are the words ending with the sequence “am” or “ans”, in which case the word is not accentuated and is a paroxytone).

As regards sequences of two or more connected vowels, it was first necessary to decide if they are on the same or different syllables, which is equivalent to determine if a vowel grapheme is a vowel or a semi-vowel. This is done by considering vowel precedence, that is, the fact that, in Portuguese, the vowel grapheme ‘a’ is always a vowel and, then, other vowel graphemes in the same syllable are semivowels. Using this reasoning, and knowing which diphthong sequences exist in Brazilian Portuguese, it is possible to differentiate the roles of the vowels and semivowels, which is also important to stress assignment. For example, in the word ‘paiol’, which we know is an oxytone, ‘o’ is the stressed vowel and, because 'a' has precedence over ‘i’, when converting to a phoneme, 'i' will be the semivowel forming a decreasing diphthong.

After the phase of stress position detection, the conversion rules are applied. They can be divided into three main categories:

C. *Direct Conversions (context independent)*

These types of conversions are independent of the context in which the grapheme is inserted, meaning that it depends solely on the grapheme or graphemes being analysed. Some examples of this type of conversion (grapheme → phoneme in ASCII) are:

b → b
f → f
j → zh ([ʒ], in the IPA representation).

D. *Word-Dependent Conversions*

These conversions depend on some contextual factor in the word (position of the grapheme, preceding or following grapheme). In order to make these conversions the whole word is needed as a context. The main factors that were taken into account were:

Position:

Some graphemes are pronounced differently depending on the position in the word, mostly the end or the beginning of the word or depending on their positions in the syllable. Syllable boundaries can be recognized easily for certain consonant sequences such as /lC/, /RC/, /sC/ and then, rules referring to syllable boundaries can be applied, such as the following:

‘l’ is converted to the symbol "U" ([ʊ] as the IPA equivalent) when in the end of the syllable;

‘r’ in the end of a syllable is converted to the symbol "R" as an archiphoneme, without the specification of the specific pronunciation, which is highly variable in Brazilian Portuguese [9][19], whereas in the beginning of the word it is converted to the symbol "r";

‘s’ in the beginning of the word is converted to the symbol "s";

‘s’ in the end of a syllable is converted to the symbol "S" (as in a archiphoneme) for differentiating the realization of /s/ in coda from that in onset position, as above.

Furthermore, vowels before stress are represented with lowercase (e.g., akamadU, for "acamado" and with uppercase after stress (e.g., kazA, for "casa"), not only following the proposal by [1], but mainly for allowing direct use in Praat scripts for phonetic analysis.

Preceding and/or Following Letter:

Some graphemes are pronounced differently depending on the letter that precedes or follows them. This can be seen in the examples below:

‘s’ when between vowels is pronounced as [z];

‘s’ when between followed by ‘s’ is pronounced as [s];

‘c’ when followed by ‘h’ is pronounced as [ʃ] ([sh] in the ASCII representation), whereas it is pronounced as [s] after "e, i" and [k] after "a, o, u";

‘u’ is not pronounced when in the sequences ‘qui’, ‘que’, ‘gui’ and ‘gue’ (exceptions are included in the exceptions dictionary, like for "arguição" and "água", see below).

E. Part-of-speech-Dependent Conversions

These conversions depend on the part of speech of the word in which the grapheme is inserted (if it is a verb, adjective, etc). Some examples are:

jogo N (game) [o] vs. jogo V [ɔ] (I play)
olho N (eye) [o] vs. olho V [ɔ] (I look)

Part-of-speech was obtained by using the lemmatizer conceived by the team of researchers of the Núcleo Interinstitucional de Linguística Computacional of the University of São Paulo and implemented by [17]. It is based on the MXPOST part of speech tagger and UNITEX dictionaries for Portuguese delivering the lemmas and Part-Of-Speech tags of the words of a text stored in a plain text file. Due to the slowing down of the AlinhaPB performance on the web page, the use of lemmatizer was suspended for further evaluation and, for the time being, the most frequent pronunciation of the aperture degree of the medial vowels is being used by the converter.

Rules depending on the semantic value of the word, as in the case of "sede" (thirst) [e] vs. "sede" [ɛ] (headquarters), were not implemented for lack of a semantic parser. In that case, the most frequent conversion was chosen as a default. Word frequency can be checked by a simple search of the list of words in the Corpus Brasileiro database [4].

F. Interword rules (sandhi)

These conversions are a special case when a word ends in an unstressed ‘a’ and the following word begins with an unstressed vowel, either identical or not [5]. In this case, the ‘a’ is not pronounced. Below we can see one example in which it happens and one in which it does not:

‘linda armada’ → "liNdaRmadA"
‘linda árvore’ → "liNdA aRvORI"
'a bela irmã' → "a behliRmaN"

Because sandhi has its phonetic implementation more variable when the word ends with unstressed 'e', 'i', 'o 'and 'u', such as in "o belo irmão", where the final vowel of the first word being can either be realized as a semivowel or can be elided, no rule was implemented for these cases [6].

G. Exceptions

To handle exceptions to the rules above, a dictionary was created so that words that are in this dictionary have their complete phonetic conversion ready, thus avoiding the need to use the rules and therefore dodging these mistakes. Only words to which it wasn’t possible the creation of a rule were added to the dictionary. Some examples of these: ‘pinguim’, ‘linguiça’.

If the word still has the trema over "u" (that is, "ü" like in "lingüística"), following the previous orthographic rules, it will always be converted to "U". In the other cases, where the grapheme "u" is followed by a vowel, the only way to determine in which words the ‘u’ is pronounced as a semivowel is by means of an exceptions dictionary. Because the "u" in these conditions (e.g., "quilo", kilo, "guerra", war, "aquele", that) is not pronounced in the large majority of the times, the cases in which it is pronounced were added to the exceptions dictionary;

‘logo’ [o] (noun) and ‘logo’ [ɔ] (verb), and other homographs.

The pronunciation of the vowels in these words can only de determined by their part of speech or semantic value inferred from the context and these cases were explained in section E.

H. Grouping the phonemes/allophones into words and syllable-sized units

Splitting the converted text into words and single phonemes/allophones is straightforward, consisting of simply separating the text after every white space or after every phoneme/allophone, respectively. But assembling the phonemes into phonetic syllables requires some processing. For the purpose of this project, phonetic syllables consist of units of one or more segments starting with a nucleus formed by the vowel or a diphthong and followed by semi vowels or consonants to form the so-called VV unit [2].

By using a simple correspondence table, we can divide the converted string of phones/phonemes into vowels and consonants. Because a VV unit must start with a vowel, all following semivowels and consonants are placed at the end of this unit. Furthermore, a vowel grapheme can follow a stressed vowel in the syllable, but a stressed vowel cannot follow a vowel grapheme and being in the same syllable (e.g., in the word "saída", exit, the two first vowels are in different syllables, whereas in the word "sai", s/he goes out, the two vowel graphemes are in the same syllable and the second one is a semivowel). Using these rules and disregarding the spaces dividing the words, we can determine the VV units as either a vowel followed by one or more consonants or semivowels or with a single vowel for the case it is not followed by consonants (e.g., 'a árvore', the tree). The only exception that must be made is that, if the sentence starts with a consonant, this consonant will be isolated, as there is no vowel preceding it. An example of this is shown below:

‘Mas que seja...’ → ‘ m aSk es ezh A...’

I. *Workflow*

The path for finding the phoneme/allophone conversion of an input text is shown in the flowcharts below, where G2P means Grapheme-to-Phoneme conversion. Fig. 1 presents a macroview of the conversion procedure and Fig. 2 presents a detailed view of the same procedure.

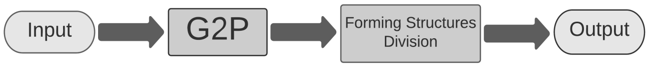

Fig. 1. Flowchart describing the process of conversion of an input grapheme string to a phoneme string (output).

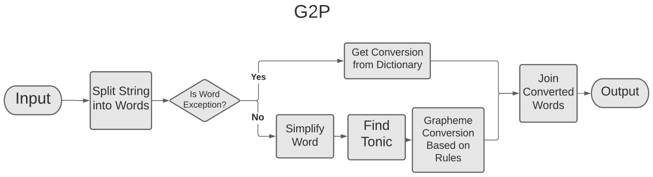

Fig. 2. In-depth flowchart of the grapheme-to-phoneme converter.

J. *Conversion results*

The converter was tested with a selection of texts in order to have as many possible variations for each phoneme/allophone and validated against hand-made conversions. In total, a set of 1,500 phonemes were used. The converter had an error of less than 0.1% for this set, only making four mistakes. The main source of mistakes was in the differentiation of the open and closed sound for the vowels ‘o’ and ‘e’, as no rule that adequately describes the differentiation was found. On the other hand, all these errors in specific words can be added to the exception dictionary, so that they are corrected in the next conversion.

III. Aligner

The aligner was responsible for receiving an audio file and a phoneme-converted text at the input and generate the corresponding time values in the audio that correspond to the boundaries of each phoneme/allophone (phonetic syllable or word) obtained from the previous analysis. Three approaches were used for training, one using HTK directly, one using Prosodylab-Aligner (which also uses HTK) and the last one using Montreal Forced Aligner (which uses the Kaldi technique). For each one of these approaches, a model was trained for each phoneme/allophone (and silence) in both the training and decoding phases.

A. *Data*

For the first training, annotated audio files for six different speakers were used, where the audios were delimited in phonetic syllables (VV units) and labeled using the ASCII symbols of Tab. I. The speakers were all from the State of São Paulo, aged from 20 to 35 years, being five men and one woman studying at the University of Campinas at the undergraduate or Graduate levels. Three additional male speakers were spared for the validation of the model on speakers that were not used in training, to see how well it would perform. In total, 363 audios and 4,117 phonetic syllables (circa 8,900 phonemes) were used for training the models. The audio files, of high acoustic quality, were formed by excerpts of reading and storytelling from the Belém corpus, which consists of the reading of a 1,600-word story of the origin of the Belém pastries in Portugal and subsequent telling of the story, Belém dataset (see [3] for the use of this corpus to compare the rhythm of Brazilian and European Portuguese). Two other datasets with no environment control were added in order to evaluate the impact of noisy audio files on model performance. They were Vox dataset with at least 89 speakers with a total of 73,025 phonemes, and LapsBM dataset with 35 speakers (10 women), each one with 20 single utterances, totaling 700 utterances. This also allowed including more female speakers to modeling. Both datasets are available at <https://github.com/igormq/asr-study/tree/master/datasets>. They do not have the corresponding phoneme alignment, which was supplied by the output of the models trained with the high acoustic quality dataset.

B. *Training*

Due to restrictions inherent to the Belém dataset, originally segmented into phonetic syllables, the first task was to split these syllables into their constituent segments for at least two reasons: (1) because the aligner can only align phonetic material for which it has a correspondent model and the number of possible phonetic syllables is far higher than the number of phonemes/allophones (exponentially larger), the creation of an appropriate model for each syllable is impracticable, as the training dataset would not have all the possible phonetic syllables in a sufficient number for an accurate training, due to the lack of enough examples for each phonetic syllable type; (2) by training the model on phonemes/allophones, the number of training tokens for each phoneme is much larger than the number we would have for each phonetic syllable. Those are the reasons why the training was made using phonemes/allophones. For that, a correspondence table was used, so that each syllable was divided in its constituent phonemes/allophones. The following models were built based on the available toolkits.

The first training used HTK directly and converged in seven iterations with the insertion of models for a silence interval, so that each syllable would have a silent acoustic model attached to it (in the case the syllable is nor preceded or followed by a silent pause, silence duration was set to 0). It is also worth pointing out that each phoneme was represented by a five-state model states (three emission states and the input/output states) including silence. By using this, it was possible to obtain a model that considered the spectral changes through time for each acoustic segment. Models with both monophones and triphones were used.

For the Prosodylab-Aligner (PL) training, ten iterations were used per round with all the default configurations. These configurations include three rounds of training and 16 kHz for the sample rate (which is the recommended sample rate because of the fact that it is the default one in this case). This default required that the entire dataset was resampled to 16 kHz, because the original sampling rate was 22,05 kHz.

Finally, for the Montreal Forced Aligner (MFA), the default configurations were used, that is, 39 iterations for the phoneme/allophone models (where each phoneme/allophone is modeled in the same way, regardless of phonological context) followed by 40 iterations for the triphone training (where the immediate context on either side of a phone is taken into account for the acoustic models) and a third and forth passes that find a transform that makes the phonemes/allophones maximally different and enhance the triphone models taking the speakers into account, respectively. It is worth noting that the triphone pass is different than training over phonetic syllables in the sense that it takes the monophone models trained in the previous step and now considers the context of the monophones in each side for a more robust and context-dependent training. Only the MFA acoustic models considered triphones, because PL does not offer this option, and, for both the PL and MFA models the audios were resampled to 16 kHz. As the training algorithm is not guaranteed to achieve a global maximum, several iterations were made of the models in all the training options and the best one, that is, the one with the least average error, was chosen as the final result. It must be said that, even with the additional triphone training for MFA, it did not perform better than PL.

Training was also done with the Vox and LapsBM datasets, which are, depending on the file, very noisy.

C. Alignment

For the alignment as a whole, the orthographic text and the corresponding audio file represent the input. The text is then converted to its phoneme/phone representation. Finally, the phoneme/allophone labels are then organized in the desired alignment structure (phonemes/allophones, phonetic syllables or words). The alignment is then made using the Viterbi decoder in HTK using the single phoneme models and a dictionary that relates the desired alignment structure and the phonemes/allophones of the acoustic models.

D. Validation

For the validation, the absolute difference between the aligned interval duration and the manually annotated interval duration was used for probing the models. As aforementioned, the validation data was split between training and test sets with different speakers.

IV. Results

Comparing the results between aligners for the Belém training set and in Tab. II, we can see that both the PL and the model trained with HTK toolkit, monophones, performed very similarly for the speakers present in the dataset, obtaining over 85% of the alignments with less than 25 ms and mean errors of 19 ms and 21 ms, respectively. These results are exactly of the same size of the recently developed Kaldi-based aligner for Brazilian Portuguese by [11], which required much more costly training phase with more than 170 hours of audio data. See also those developed by [10][13].

TABLE II: Aligner absolute errors for speakers present in the training dataset (Belém dataset)

Cumulative Error	HTK	Prosodylab Aligner	Montreal Forced Aligner
< 10 ms	47.1%	38.5%	44.6%
< 25 ms	85.5%	85.2%	73.51%
< 50 ms	93.5%	94.1%	85.4%
< 100 ms	96.5%	98.3%	92.6%
< 200 ms	98.8%	99.8%	97.7%
Mean	21 ms	19 ms	33 ms
Median	11 ms	13 ms	12 ms
Standard Deviation	50 ms	41 ms	95 ms

We can see in Fig. 3 that the predicted and manual alignments matched very well for one example of the alignment of an audio of a speaker in the training set performed with the PL toolkit. Observe that both labels and time alignment match almost perfectly, with very small displacements. The MFA also managed to perform fairly well, achieving over 73% of the alignments with less than 25 ms of error and over 97% with less than 200 ms. As we can see, it did not perform quite as well as the previous aligners, with a mean error more than 10 ms higher than the others. Even so, its median only differed by 1 ms from the other aligners, which is due to the fact that the MFA error distribution was more spread out, as can be seen in the histogram in Fig. 4, which can be compared with the one in Fig. 5, for speakers not present in the training set, where one can see that the distributions are quite similar, confirming the relative speaker independence of this toolkit.

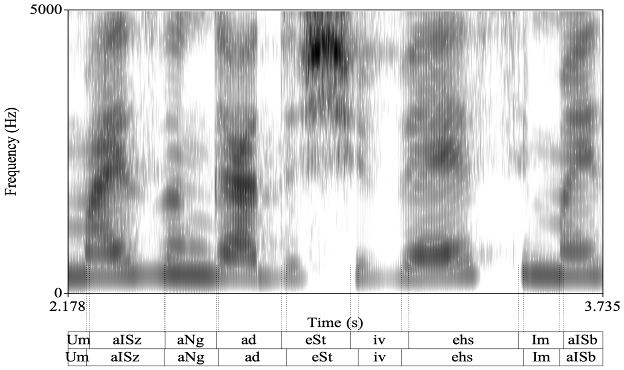

Fig. 3. Broadband spectrogram (above) and corresponding original (bottom) and predicted (medial) VV labels and boundaries for a speaker in the training set for the excerpt "(quant)o mais zangado estivesse, mais b(aixava a voz)".

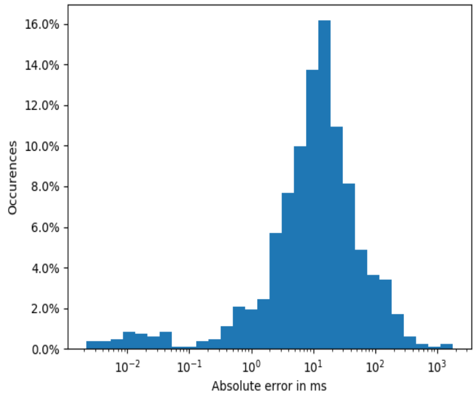

Fig. 4. Percentage of time errors in milliseconds for speakers present in the training set plotted in a log scale for the models trained directly with Montreal Forced Aligner

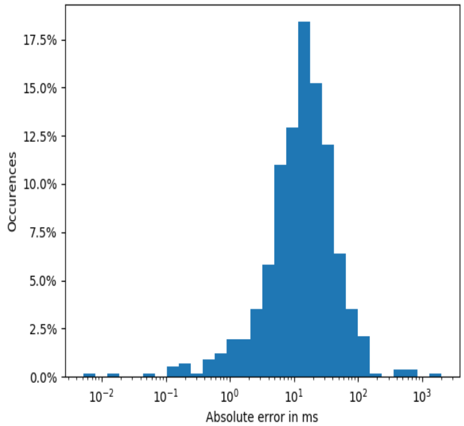

Fig. 5. Percentage of time errors in milliseconds for speakers not present in the training set plotted in a log scale for the models trained directly with Montreal Forced Aligner

Although the models performed well for the tests in the Belém training dataset, another validation with speakers who were not in the training dataset was made for the other two toolkits as Hidden Markov Models are known to be speaker-dependent. In this other validation, we can see in Tab. III that the model trained directly with HTK had a large decrease in performance, having only 30.4% of the errors below 25 ms (a drop of 55.1% in comparison to speakers in the training dataset). The MFA and PL toolkits, on the other hand, maintain the error rates in the test dataset, obtaining 72.2% and 87.8% respectively for their errors below 25 ms and mean errors of 16 ms and 29 ms. PL performed slightly better than MFA for speakers not seen during training may be due to the fact that the audios being used were not phonetically balanced, and the particularities of both the training and test sets and the training process itself may end up resulting in this difference. The PL toolkit ended up having results comparable to the MFA-LA model trained on the Buckeye dataset presented here [18], which had a mean error of 17 ms and a median of 11.2 ms (although it is worth pointing out that the MFA-LA model aligned phonemes and not phonetic syllables).

TABLE III: Aligner errors for speakers present in the test dataset (Belém dataset)

Cumulative Error	HTK	Prosodylab Aligner	Montreal Forced Aligner
< 10 ms	12.8%	41.4%	35.6%
< 25 ms	30.4%	87.8%	72.2%
< 50 ms	53.5%	97.5%	90.3%
< 100 ms	88.9%	99.3%	96.8%
< 200 ms	96.6%	99.6%	99.1%
Mean	63 ms	16 ms	29 ms
Median	46 ms	12 ms	15 ms
Standard Deviation	93 ms	34 ms	101 ms

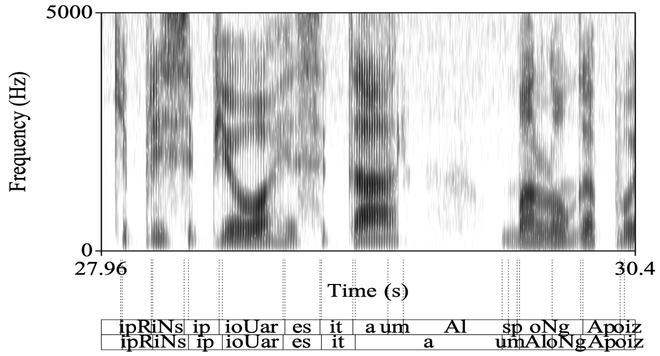

Fig. 6. Broadband spectrogram (above) and corresponding original (bottom) and predicted (medial) VV labels and boundaries for a speaker not in the training set for the excerpt "e principiou a recitar uma longa poes(ia)".

As can be seen in the example of Fig. 6 for the PL model, the predicted alignments match the manual annotation quite faithfully with the main differences being the incorrect attribution of (1) the VV label "Al" to a silent pause, (2) the VV label "um" to the end of the [a] of "recitar", and (3) the silent pause label (sp) to the "m" of the word "uma", where segment /u/ was not pronounced. Note that the VV unit [a] of the final vowel of "recitar", pronounced as "recitá", a common pronunciation for the verbal infinitive, includes the silence pause, as required from the definition of this unit.

Figs. 7 to 10 compare the distribution of errors for HTK and PA aligners, for both the Belém training and test datasets. As it is shown in the in Fig. 7 vs. 8, and Fig. 9 vs. 10, the errors for PA were less spread out, being more centered around their mean than the ones for the model trained directly with HTK. We can also see that, especially for the PA alignments, the curve is left skewed, as over 90% of the errors are below 50 ms in both tests. In the HTK model, on the other hand, the histograms are more symmetrical, particularly for the speakers in the test dataset (50% of the errors below 50 ms).

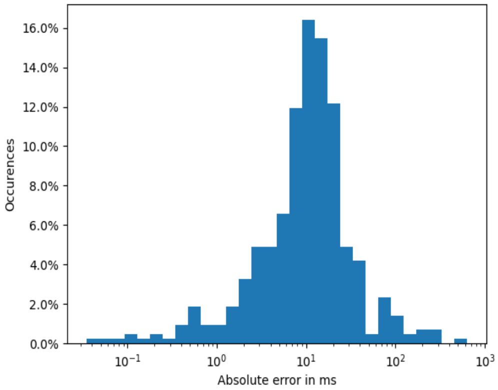

Fig. 7. Percentage of time errors in milliseconds for speakers present in the training set plotted in a log scale for the models trained directly with HTK.

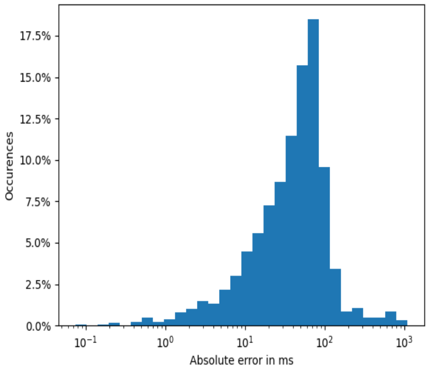

Fig. 8. Percentage of time errors in milliseconds for speakers not present in the training set plotted in a log scale for the models trained directly with HTK.

Even though the means and medians of the time errors in the models are close for the training and test datasets, particularly for the PA toolkit, the plots show that the three toolkits have some alignments with particularly high errors (on the order of 100 ms for the Prosodylab and 1000 ms for the HTK), which could be extremely harmful if their frequency were not so small (well below 0.5%).

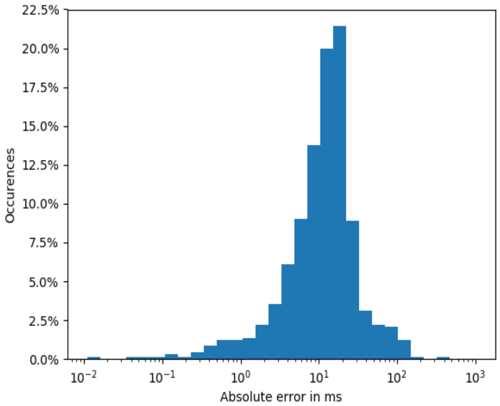

Fig. 9. Percentage of time errors in milliseconds for speakers present in the training set plotted in a log scale for the models trained directly with Prosodylab-Aligner.

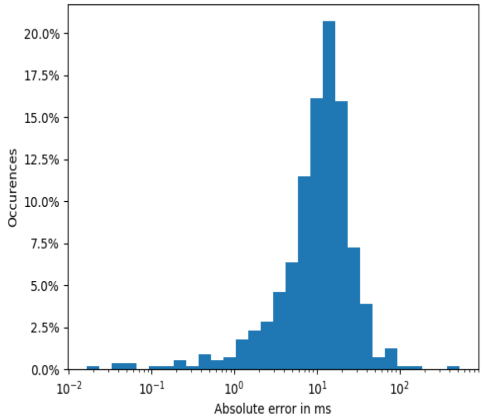

Fig. 10. Percentage of time errors in milliseconds for speakers not present in the training set plotted in a log scale for the models trained directly with Prosodylab-Aligner.

The difference between MFA and PL could be due to the fact that the MFA training involved triphone models, while the PL did not, and perhaps the dataset was not large enough to allow any reliable training of the models with enough variations of triphones. The training with the (much larger) LapsBM dataset slightly improved the results for MFA, but not PA. Another option that could be explored was to use new data to adapt the trained models, which might be possible both in the MFA and HTK cases for example.

Training with the three datasets for the MFA toolkit revealed results very close to the ones in Table II. They improved to 80% of errors below 25 ms, and 91% below 50 ms when the Vox dataset, the noisier one, is excluded. Using triphones with the HTK toolkit did not produced results better than with monophones: 74% of errors below 25 ms, and 85% below 50 ms (Belém dataset only), and 78% of errors below 25 ms, and 85% below 50 ms (Belém and LapsBM datasets).

V. Conclusion

Alinha-PB offers a base ground so that phoneticians do not have to align everything by hand, but just correct the mistakes of the forced aligner, which can drastically speed up the process. As described below, Alinha-PB offers a tool to align Brazilian Portuguese audios without the need to have a huge dataset to train your own models (even though it is still possible). The website allows making the alignment on site without the hassle to download and install new packages and software for it. Lastly, as the code is open source, it can still be improved and adapted for specific use cases by the end user, although this is not necessary for the basic use.

Compared with the recently developed aligner for BP by Dias and colleagues (2020), ours has the advantage of as faster training phase based on much less audio material, and an immediate availability in the Internet.

To make easier the use of the developed aligner and converter, a website was created (<https://conversoralinhador.herokuapp.com/>), so that the user wouldn't have to download anything to use them. For the converter, the user has the options of converting the grapheme text and tuning the exceptions. The latter is offered so that if a word is incorrectly translated, you can add or remove the exceptions in the exception dictionary so that the conversion will be corrected. For the aligner, the user has to upload a wav audio file and provide the corresponding text (it can be given in phonemes or graphemes, in which case the converter would be responsible for converting the text to phonemes/allophones) and can choose the alignment (between phonemes, phonetic syllables and words or the three altogether). As was shown above that the aligner works better for speakers that were used in the training dataset, an option to train a new model with pairs of audio and the corresponding text files with the correct transcriptions is provided. These trained models can be created using tools such as Prosodylab-Aligner, which are free to use and only require the audios and their respective transcriptions in phonemes (the transcription does not need to be time-aligned). All the trained models and a local version of the website are also available here <https://github.com/jkruse27/Alinha-PB>.

REFERENCES

[1] E. C. Albano and A. A. Moreira, "Archisegment-based letter-to-phone conversion for concatenative speech synthesis in Portuguese," in *Proc. of Fourth International Conference on Spoken Language Processing*, Philadelphia, USA, 1996, pp. 1708-1711.

[2] P. A. Barbosa, *Incursões em torno do ritmo da fala*, Campinas, Brazil: Pontes, 2006.

[3] P. A. Barbosa and W. da Silva, "A New Methodology for Comparing Speech Rhythm Structure between Utterances: Beyond Typological Approaches, " in *PROPOR 2012*, LNAI 7243, H. Caseli et al., Eds. Springer: Heidelberg, 2012, pp. 329-337. DOI: 10.1007/978-3-642-28885-2_37.

[4] T. Berber Sardinha, J. L. Moreira Filho, and E. Alambert, "O Corpus Brasileiro," presented at the VII Encontro de Lingüística de Corpus, Unesp, São José do Rio Preto, Brazil, November 6-7. Available: <http://corpusbrasileiro.pucsp.br>, 2008.

[5] L. Bisol, "Sândi Vocálico Externo: Degeminação e Elisão," *Cadernos de Estudos Linguísticos,* vol. 23, pp. 83–101, 1992. DOI: 10.20396/cel.v23i0.8636847.

[6] L. Bisol, "Sandhi in Brazilian Portuguese," *Probus*, vol. 15, no. 2, pp. 177-200, 2003. DOI: 10.1515/prbs.2003.007.

[7] L. Bisol, "O acento: duas alternativas de análise," *Organon*, vol. 28, no. 54, 2013. DOI: 10.22456/2238-8915.41192.

[8] P. Boersma and D. Weenink, *Praat: doing phonetics by computer* [Computer program]. Version 6.1.26 Available: http://www.praat.org/, 2020.

[9] D. M. I Callou, J. A. de Moraes, and Y. Leite, "Variação e diferenciação dialetal: a pronúncia do /r/ no português do Brasil," in *Gramática do português falado*, vol. 6, I. Koch, Org. Campinas, Brazil: Editora da Unicamp, 1996, pp. 465-493.

[10] L. G. D. Cuozzo et al. "CNN-based phonetic segmentation refinement with a cross-speaker setup." in Int. Conf. on Computational Processing of the Portuguese Language. Springer, Cham, 2018. p. 448-456. DOI: 10.1007/978-3-319-99722-3_45.

[11] A. L. Dias, C. Batista, D. Santana, and N. Neto, "Towards a Free, Forced Phonetic Aligner for Brazilian Portuguese Using Kaldi Tools," in *Proc. Brazilian Conference on Intelligent Systems,* Cham, 2020, pp. 621-635. DOI: 10.1007/978-3-030-61377-8_44.

[12] M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition," *Found. Trends Signal Process*, vol. 1, no. 3, pp. 195-304, 2007. DOI: 10.1561/2000000004.

[13] J.-Ph. Goldman, "EasyAlign: an automatic phonetic alignment tool under Praat," in Proc. of InterSpeech 2011, Firenze, Italy. Available at <http://latlcui.unige.ch/phonetique/easyalign.php >.

[14] K. Gorman, J. Howell, and M. Wagner, "Prosodylab-Aligner: A Tool for Forced Alignment of Laboratory Speech," *Canadian Acoustics*, vol. 39, no. 3, pp. 192–193, 2011. Available: <https://jcaa.caa-aca.ca/index.php/jcaa/article/view/2476>

[15] G. E. Hinton, and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, no. 313, pp. 504–505, 2006. DOI: 10.1126/science.1127647.

[16] S. Hunnicutt, "Grapheme-to-Phoneme rules: a Review," Speech Transmission Laboratory, Royal Institute of Technology, Stockholm, Sweden, QPSR, no. 2-3, pp. 38-60, 1980.

[17] E. G. Maziero, "Lemmatizer for Portuguese," Available: <https://sites.icmc.usp.br/taspardo/LematizadorV2a.rar>, 2012.

[18] M. Mcauliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: trainable text-speech alignment using Kaldi," in *Proc. of the 18th Conference of the International Speech Communication Association*, 2017. Available: <http://montrealcorpustools.github.io/Montreal-Forced-Aligner/>

[19] R. B. Mendes, "Sounding Paulistano: Variation and Correlation in São Paulo," presented at the NWAV39, San Antonio, Texas, USA, 2010.

[20] D. Povey et. al., "The Kaldi Speech Recognition Toolkit," presented at the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, 2011.

[21] U. D. Reichel, "PermA and Balloon: Tools for string alignment and text processing, " in *Proc. Interspeech 2012*, Portland, Oregon, USA, 2012, paper no. 346. DOI: 10.5282/ubm/epub.18042.

[22] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book*. Microsoft Corporation and Cambridge University Engineering Department, 2001.

João G. S. Kruse was born in 2000 in Brazil. He is a computer engineering student at the State University of Campinas and currently works at the Applied Science Division at the National Center for Research in Energy and Materials. His research interests are in the areas of machine learning, natural language processing, robotics, quantum computing.

Plínio A. Barbosa was born in 1966 in Brazil. He obtained his PhD degree in 1994 from the INP de Grenoble in France and his tenure in 2006 from the State University of Campinas, where he is associate professor at the Dep. of Linguistics. He is responsible for the Speech Prosody Studies Group and his interests are direct to experimental prosody, acoustic phonetics, speech rhythm, dynamical systems theory, speech pathology analysis, L2/FL phonetics. He is Editor of the Journal of Speech Sciences. Books (selection): Prosódia (Parábola, 2019), Manual de Fonética Acústica Experimental (with Sandra Madureira, Cortez, 2015). He is a member of the International Phonetic Association and a member of the Editorial Board of Phonetica.