Mispronunciation Detection in Children's Reading of Sentences

Proença J., Lopes C., Tjalve M., *Senior Member, IEEE*, Stolcke A., *Fellow, IEEE*, Candeias S., Perdigão F.

***Abstract*—This work proposes an approach to automatically parse children's reading of sentences by detecting word pronunciations and extra content, and to classify words as correctly or incorrectly pronounced. This approach can be directly helpful for automatic assessment of reading level or for automatic reading tutors, where a correct reading must be identified. We propose a first segmentation stage to locate candidate word pronunciations based on allowing repetitions and false starts of a word's syllables. A decoding grammar based solely on syllables allows silence to appear during a word pronunciation. At a second stage, word candidates are classified as mispronounced or not. The feature that best classifies mispronunciations is found to be the log-likelihood ratio between a free phone loop and a word spotting model in the very close vicinity of the candidate segmentation. Additional features are combined in multi-feature models to further improve classification, including: normalizations of the log-likelihood ratio, derivations from phone likelihoods, and Levenshtein distances between the correct pronunciation and recognized phonemes through two phoneme recognition approaches. Results show that most extra events were detected (close to 2% word error rate achieved) and that using automatic segmentation for mispronunciation classification approaches the performance of manual segmentation. Although the log-likelihood ratio from a spotting approach is already a good metric to classify word pronunciations, the combination of additional features provides a relative reduction of the miss rate of 18% (from 34.03% to 27.79% using manual segmentation and from 35.58% to 29.35% using automatic segmentation, at constant 5% false alarm rate).**

***Index Terms*—Speech analysis, Mispronunciation detection, Children's reading, Automatic reading annotation.**

I. Introduction

In the process of learning how to read, children can face phonological, phonic or rhythmical difficulties in reading aloud, reflecting different levels of fluency [1], [2]. Oral reading fluency depends on speed, accuracy, consistency of pace and expressiveness [2], [3]. The deviations to an appropriate reading include reading syllable by syllable, committing false starts followed by self-corrections, and severe mispronunciations of words. The wide range of possible problematic events presents a substantial challenge for computational systems that aim to detect these problems automatically.

There are several applications that could benefit from analyzing a child's sentence reading using speech recognition and disfluency detection techniques. Reading tutors, coinciding with the area of computer assisted language learning (CALL), may need to track a child's reading in real time against the written text, while incorrect pronunciations are identified and disallowed. Projects that have aimed to create automatic reading tutors include LISTEN [4], Tball [5], SPACE [6] and FLORA [7]. Another application of reading aloud analysis is automatic literacy evaluation, where reading level is computed for a child through the analysis of their reading performance [8]–[11]. Performance metrics based on reading attempts include correct words per minute and rates of reading errors. Detecting all the different types of reading events has also been useful for the automatic annotation of speech databases [12].

Although this study considers children aged 6 to 10 reading in their primary (native) language, the area of reading analysis is also relevant for second language learning. However, automatic analyses of foreign language reading [13], [14] are mostly targeted at adults or young adults for whom speech technologies are significantly more mature. Moreover, although similar reading problems may be encountered for young and adult readers, most problems in young children arise from the inability to follow phonological and phonotactic rules, as well as hesitations, self-corrections and slow reading speed. It is rarer to find problems of badly realized vowels, often the case in second language reading.

There are several known methods to detect disfluencies, such as those based on hidden Markov models (HMMs), maximum entropy models, conditional random fields [15] and

Submitted on 05-09-2017 for review. This work was supported in part by Fundação para a Ciência e Tecnologia under the project UID/EEA/50008/2013 (pluriannual funding in the scope of the LETSREAD project at Instituto de Telecomunicações). Jorge Proença is supported by FCT grant SFRH/BD/97204/2013.

J. Proença and F. Perdigão are with Insituto de Telecomunicações and the Department of Electrical and Computer Engineering of University of Coimbra, DEEC polo 2 Univ. Coimbra, PT3030-290 Coimbra, Portugal (e-mails: jproenca@co.it.pt, fp@co.it.pt).

C. Lopes is with Instituto de Telecomunicações and the Polytechnic Institute of Leiria, Morro do Lena, Alto do Vieiro, PT2411-901 Leiria, Portugal (e-mail: calopes@co.it.pt).

M. Tjalve is with Microsoft and University of Washington, Seattle, WA 98195-2425, USA (e-mail: michael.tjalve@microsoft.com).

A. Stolcke is with Microsoft AI and Research, 1020 Enterprise Way Sunnyvale, CA 94089, USA (e-mail: andreas.stolcke@microsoft.com).

S. Candeias is with Microsoft, Digital Advisory Services, Rua do Fogo de Santelmo, Lote 2.07.02, 1990-110 Lisbon, Portugal (e-mail: v-sacand@microsoft.com).

classification and regression trees [16], though most of past research focusses on spontaneous speech. Applicability to read speech may not be straightforward since disfluencies vary according to different speaking styles [17]. Disfluencies in reading have different nuances, and some prior work has targeted the automatic detection of these events in children's reading, with the most relevant contributions described below.

Black et al. [5] aim to automatically detect disfluencies in isolated word reading tasks. They found that human evaluators rated fluency as important as accuracy when judging reading ability. The target of detection is mostly sounding-outs, where a child first reads phoneme by phoneme (which can be whispered) and then reads the complete word. They build HMMs and a grammar structure specialized for disfluencies, capable of detecting partial words and allowing silence or noise between phones. The correct word is compulsorily considered to be pronounced in the final state of the grammar. They achieved 14.9% miss rate and 8.9% false alarm rate for the detection of hesitations, sound-outs, and whispering. By comparison, in our data, no phoneme-by-phoneme sounding-out was found. Instead, there are syllable-by-syllable sounding-outs with possible silence between syllables, which we will address.

Duchateau et al. [9] also target the reading of isolated words. Based on HMMs, they use a two-layer decoding module, first with phoneme decoding using phoneme-level finite state transducers to allow false starts with partial pronunciations, and then a second lattice to allow for repetitions or deletions of words. For the detection of reading errors on word reading, a miss rate of 44% and a false alarm rate of 13% were achieved. For a pseudoword reading task, they achieved a 26% rate of both misses and false alarms. In Yilmaz et al. [18], an extension to the work by Duchateau et al. [9] is developed. The new evaluation is on a mixture of word and sentence reading tasks, and the models are still based on HMMs. The decoding scheme is more flexible to allow for the most common substitutions, deletions and insertions of phones in the language, as described by a phone confusion matrix. This confusion matrix was obtained by comparing the output of the recognizer with transcripts on a larger corpus. The final result for the detection of all disfluencies (word repetitions, stuttering, skipping and mispronunciations) was 44% miss rate at a 5% false alarm rate.

Hagen et al. [19], targeting partial word pronunciations, found that syllables were the best subunits to use in a decoding lattice to detect these events. A 34.6% detection rate of partial words is achieved for a 0.5% false alarm rate, and the overall word error rate was similar to using a decoding grammar based solely on words.

Li et al. [20] aim to track children's reading of short stories for a reading tutor. As a language model, they employ a word level context-free grammar for sentences to allow some freedom in decoding words. Each word also had a concurrent garbage model with the most common 1600 words, which allows detecting word level miscues, but was also able to detect some sub-word level miscues for short words. On a detection task of all reading miscues (including breaths and pauses), they achieved a miss rate 23.07% at a false alarm rate of 15.15%.

It should be mentioned that much of the prior research focuses on individual word reading tasks – exceptions being [20] and parts of [18] –, whereas the present work targets the reading of sentences and pseudowords. Some studies go further and attempt to provide an overall reading ability index that should be well correlated with the opinion of expert evaluators [8], [9]. Overall reading assessment is also a direct application of our work.

With the objective of automatically detecting the most common reading miscues in sentences, focusing on mispronunciations, we propose a first segmentation stage to detect candidate word pronunciations, while allowing word repetitions and false starts based on syllable units. A decoding grammar based solely on syllables allows silence to appear during a word pronunciation, addressing the problem of intra-word pauses. At the second stage, candidate segments are classified as mispronounced or not by using several proposed features, with the main one being a log-likelihood ratio between a free phone loop and a word spotting model in the very close vicinity of the candidate segmentation. We combine additional features (normalizations of the log-likelihood ratio, features derived from the likelihoods of individual phonemes, and Levenshtein distance between the correct pronunciation and sequence of recognized phonemes) in multi-feature models to further improve mispronunciation classification. Although we call the second stage mispronunciation classification, it is in essence a detection task, since candidate segments must be detected automatically (as attempted in the first stage). By using segmentation information from a manual transcription, it is more clearly a case of classification.

Although incorrect intonations of a word may relate to incorrect reading, we will only focus on deviations from the ideal phonetic pronunciation, which are those given by the manual transcriptions. In fact, the features we used are derived from speech recognition/decoding paradigms. Correct prosody is linked to a good reading performance and we only partially address it by considering duration metrics. We have shown in previous work [10], [11] that considering different types of disfluency rates in addition to reading speed features (without other prosody metrics) can already improve the prediction of a child's overall reading level (from 0.92 correlation for correct words per minute to 0.95 using multiple features). The output of the developed methods is also a full automatic annotation of children's read utterances.

Compared to our previous approaches [12], [21], we propose here a new decoding strategy using only syllables in a semi-constrained way. We also use and define multiple features for mispronunciation classification, which we believe is novel in this scope.

Section 2 presents the dataset of European Portuguese children reading sentences and pseudowords that we used, characterizing the types and frequencies of disfluencies encountered. Section 3 presents the two-stage methodology of automatically detecting disfluencies: segmentation to obtain candidate units and consequent mispronunciation classification. Section 4 presents results and discussion, including a description of the metrics used.

II. Data of Children Reading

To better contextualize the problem of automatic recognition of children reading, the data used throughout this study is presented first. We chose a subset of the LetsRead corpus of European Portuguese children reading aloud [22]. The material of the reading tasks given to primary school children (6-10 years old) include sentences and lists of pseudowords, recorded in primary school classrooms with low noise and low reverberation.

The sentences present varying length and difficulty, increasing on average for higher school grades. They were extracted from children's tales and school books at the level of the target group. Pseudowords (such as <traba> [tɾˈabɐ], <impemba> [ĩpˈẽbɐ] or <culenes> [kulˈɛnəʃ][1]) represent non-existing or nonsense words in the native language, used to evaluate morphological and phonotactic awareness. Pseudowords of 2 to 4 syllables were created by shuffling the most common syllables in a lexicon of European Portuguese, maintaining full pronounceability [22].

TABLE I. Frequency and description of disfluency types. Relative values are over the total number of prompt words (39826).

Tags	Frequency	Description
PRE	1586 (4.0%)	False starts that are followed by the attempted correction (pre-corrections), where multiple can occur. Example: for prompt "*grande espanto*" [gɾˈɐ̃də (i)ʃpˈɐ̃tu], utterance is "*grande* ***espa*** *espanto*" [gɾˈɐ̃də **ˈiʃpɐ** iʃˈpɐ̃tu].
SUB	1396 (3.5%)	Substitution or severe mispronunciation of a word. Example: for prompt "*voava em largos círculos*" [vuˈavɐ ɐ̃j̃ lˈaɾguʃ sˈiɾkuluʃ], utterance is "*voava em* ***lares sicos***" [vuˈavɐ ɐ̃j̃ **lˈaɾəʃ sˈikuʃ**].
PHO	2124 (5.3%)	Small mispronunciation of a word, usually with a change in one phone. Example: for prompt "*A Lena chegou a casa, da escola*" [ɐ lˈenɐ ʃəgˈo ɐ kˈazɐ dɐ (i)ʃkˈɔ.lɐ], utterance is "*A Lena* ***chegou*** *a casa, da* ***escola***" [ɐ lˈenɐ **ʃigˈo** ɐ kˈazɐ dɐ **ɛʃkˈɔlɐ**].
REP	776 (1.9%)	Repetition of a word (multiple repetitions may occur). Example: for prompt "*Ele já me deu*" [ˈelə ʒa mə dew], utterance is "*Ele,* ***ele*** *já me deu*" [ˈelə **ˈelə** ʒa mə dew].
INS	265 (0.7%)	An inserted word that is not part of the original sentence. Example: for prompt "*mas também dizem*" [mɐʃ tɐ̃bˈɐ̃j̃ dˈizɐ̃j̃], utterance is "*mas também* ***me*** *dizem*" [mɐʃ tɐ̃bˈɐ̃j̃ **mə** dˈizɐ̃j̃].
DEL	104 (0.3%)	The word was not pronounced (deletion). Example: for prompt "*onde morava* ***uma*** *velha"* [ˈõdə muɾˈavɐ **ˈumɐ** vˈɛʎɐ], utterance is "*onde morava velha*" [ˈõdə muɾˈavɐ vˈɛʎɐ].
EXT	732 (1.8%)	Extension [:], when a phone is severely extended. May occur simultaneously with other disfluency events. Example: for prompt "***ele*** *chegou a* ***casa***" [ˈelə ʃəgˈo ɐ kˈazɐ], utterance is "***e:le*** *chegou a* ***ca:sa***" [ˈe:lə ʃəgˈo ɐ kˈa:zɐ].
IWP	1336 (3.4%)	Intra-word pause […], when a word is pronounced syllable by syllable with intervening silences. May occur simultaneously with other disfluency events. Example: for prompt "*formosa e bonitinha*" [fuɾmˈɔzɐ i bunitˈiɲɐ], utterance is "*formosa e* ***boni...tinha***" [fuɾmˈɔzɐ i **buni...tˈiɲɐ**].

The manually transcribed data used in this study includes sentences and pseudowords from 213 children totaling 10.5 hours. The children are approximately equally distributed over four grade levels. This data was manually annotated in terms of correct words, mispronunciations and other disfluency events. Table I presents the frequencies and descriptions of the different types of disfluencies encountered.

Both SUB and PHO correspond to mispronunciations and, although their severity degrees are different, the two will be considered jointly as the mispronounced class. The training set used to train acoustic models for word and phone recognizers and to optimize classifiers comprises 9 hours. The remaining 1.5 hours from separate speakers are used as an independent test set.

III. Detection of Disfluencies

Automatically detecting all the different types of disfluencies encountered in children's reading has proved to be a significant challenge. In this work, the most frequent types are targeted for automatic detection – mispronunciations, false starts and repetitions – as well as intra-word pauses that can occur simultaneously with other events. It could be argued that detecting only correct words is enough to characterize children's reading in most applications, for example, to calculate correct words per minute. In that case, an approach such as word spotting could be enough to detect correct words, although it is expensive, especially for large sentences. However, it has been found that the relative amount of specific disfluencies may also be a relevant parameter for reading level assessment [10] and the strategy presented here stems from that goal.

The workflow of disfluency detection with the final goal of detecting mispronunciations follows the schematic of Fig. 1.

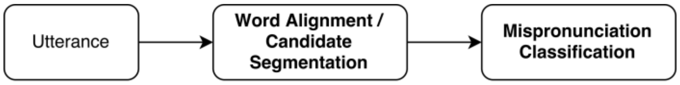
Utterance
Word Alignment / Candidate Segmentation
Mispronunciation Classification

Fig. 1. Schematic of disfluency detection workflow. The segmentation stage, while allowing extra content, provides candidate segments for words to be classified as mispronounced or not.

First, we found it necessary to obtain alignment information for any word attempts. This segmentation allows for extra content for repetitions and false starts. A candidate word segment is ideally aligned with the pronunciation attempt of its word and, through the use of several features extracted for that time-frame, may then be classified as mispronounced or not.

A. Segmentation and detection of extra events

This first stage aims to get the best alignment possible for both correct words and mispronounced attempts, with the only metainformation given being the original prompt. The first challenge for segmentation comes from all the extra content that can frequently occur (repetitions, false starts and insertions) as well as deletions. Otherwise, a forced alignment of the prompt word sequence would suffice. The second challenge is that pronunciations can differ significantly from the reference

[1] In this document, stress is marked before the vowel of the stressed syllable.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2820429, IEEE/ACM Transactions on Audio, Speech, and Language Processing

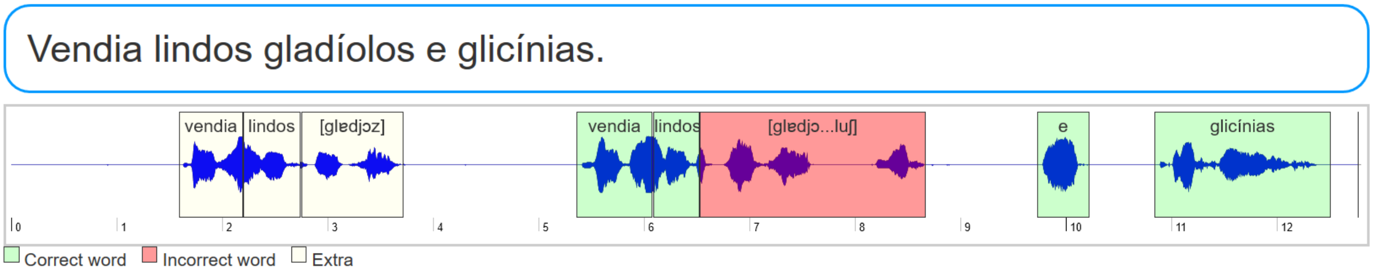

Fig. 2. Example of a prompt (top) and an utterance of its reading attempt (bottom), with segmentation for extra content and correct and incorrect words. Transcription: REP(*vendia*) REP(*lindos*) PRE([glɐdjɔz]) *vendia lindos* SUB([glɐdjɔ...luʃ]) *e glicínias*. Expected correct reading: [vẽdiɐ] [lĩduʃ] [glɐdiuluʃ] [i] [glisinjɐʃ].

pronunciation. Any decoding strategy must not be too unconstrained, since short words might otherwise be detected in false starts and mispronunciations. Consequently, the proposed decoding grammars are a mixture of strictly following the prompt with the added option of word repetition. It is still possible to obtain a good alignment for a mispronunciation, even if it deviates substantially from the reference word.

For acoustic models, we found greater success in this segmentation task by simply using triphone hidden Markov models of Gaussian mixture models (HMM-GMMs), rather than neural networks for triphone decoding. One possible explanation is that the amount of training data (9 hours) is not sufficient for neural network training. For this stage, standard triphone HMMs were trained with the Kaldi toolkit [23].

Another substantial challenge is the occurrence of intra-word pauses. These are more common for the lower grades and stem from reading a word syllable by syllable, with a significant pause in between. Therefore, pauses usually occur between syllables, whereas a regular word decoder or aligner is not expecting silence inside of a word. Fig. 2 shows an example of a problematic utterance: repetitions, a false start, and an incorrect word with an intra-word pause can be found.

For this stage, we present a baseline and two systems that deal with intra-word pauses before or during decoding:

1. A baseline system where intra-word pauses are not considered and a word-based decoding is used, allowing repetitions and false-starts.
2. A system based on previous work [21], cutting silent segments in the waveform before decoding with full words, similarly to the baseline.
3. A syllable-based decoding grammar, where silence is optional between all syllables.

The methods are described in the following subsections. For all methods, the allowed false starts (represented by the suffix PRE) are based on stopping a word pronunciation at syllable boundaries, which are common interruption points. For example, for a word with four syllables [syl1.syl2.syl3.syl4], the allowed pronunciations for a false start are [syl1], [syl1.syl2] and [syl1.syl2.syl3]. Allowing deletions was found to provide worse alignment results from the onset of this study. This is probably because introducing ‘skip’ arcs in the alignment lattices allows for a high degree of freedom that allows mispronounced words to be matched up incorrectly, as previously stated. Therefore, deletions are not allowed in either approach. Insertions of spoken content not related to the prompted words (including out-of-vocabulary words) are also not considered, but it is likely that candidate segments for insertions and deleted words (with an imperfect alignment) will be detected as mispronounced on the pronunciation classification stage.

1) *Baseline: word-based decoding*

With this approach, only repetitions and false starts are targeted and nothing is done to address intra-word pauses. For a given utterance, a decoding grammar (lattice) is built from the original prompt, allowing repetitions and syllable-based false starts. Decoding is performed using this lattice and HMM models.

The lattice built for a specific prompt is a finite state transducer (FST) based on the sequence of words of the original prompt. For each word, two additional elements are added to the lattice: an arc to go back after a word pronunciation, allowing for repetitions; and a self-loop arc before the word to allow multiple false starts. The arc weights of the FSTs used in this work were empirically decided. An example for the lattice built for the prompt *“ele sonhava muito”* [ˈelə suɲˈavɐ mˈũȷ̃tu] (he dreamed a lot) is shown in Fig. 3. False starts are represented by the suffix PRE and, in this example: *elePRE* can only be [e]; *sonhavaPRE* can be [su] or [su.ɲˈa]; *muitoPRE* can only be [mũȷ̃].

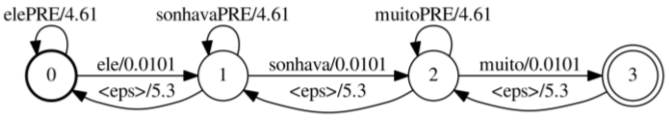

Fig. 3. Schematic of the lattice built for the prompt “ele sonhava muito”, for the word based-decoding method.

Following the horizontal left-to-right arcs, the original sentence is obtained. By following multiple arcs that transition backwards (non-consuming <eps>), repetition of sequences of words are also allowed, such as *“ele sonhava ele sonhava muito”*. These occurrences are frequent in the data and typically represent corrections initiated by repeating every word from the start of sentence or clause.

2) *Word-based decoding with silence cutting*

For this approach to word alignment based on previous work [21], silence periods are removed before decoding so that words are expected to be continuous. A neural network based on long

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2820429, IEEE/ACM Transactions on Audio, Speech, and Language Processing

temporal context [24] was also trained, targeted for phoneme recognition and achieving 27% phone error rate on the test set with a free phone-loop model. Its outputs are posterior probabilities of phones and non-speech and it is used in this method to detect non-speech segments (it will also be used for mispronunciation classification). The method follows these steps:

1. Voice activity detection. Significant non-speech segments (longer than 150ms) are cut from an utterance to deal with intra-word pauses. Non-speech segments are found from sequences that have a high probability of being silence based on the posterior probabilities output by the phonetic recognizer.
2. Decoding using task-specific grammars. For a given utterance, the same word-based decoding as described for the baseline is employed, allowing repetitions and false-starts (Fig. 3).
3. Reintroduction of non-speech segments. Finally, information pertaining to the segments of non-speech that were originally cut (either separating words or inside a word) is used to expand the decoded segmentation to the utterance's original duration.

3) *Syllable-based decoding*

For this approach, the problem of silence inside a word is handled differently. Here, the decoding strategy is based on separating a word into its syllable components and building a lattice solely with these syllables. Fig. 4 shows an example of the lattice for the same sentence "ele sonhava muito" [ˈelə suɲˈavɐ mˈũj̃tu]. The allowable repetitions and false starts are similar to the previous approach since, after a given syllable, it is only allowed to return, at most, to the beginning of the word.

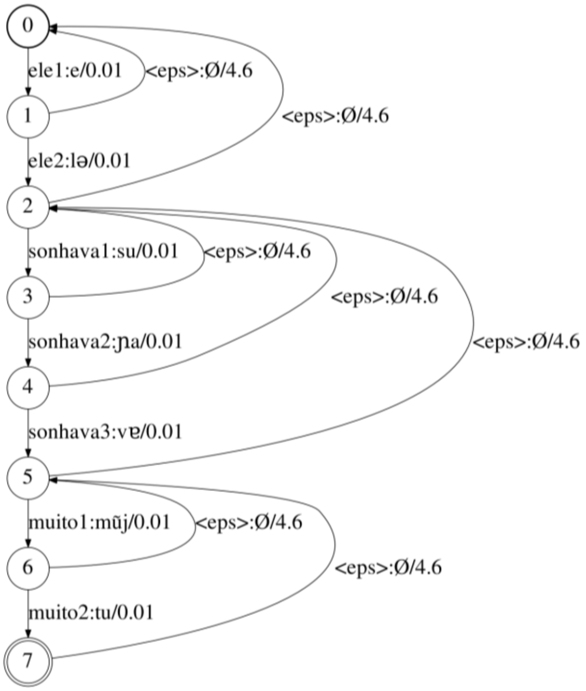

Fig. 4. Schematic of the lattice built for the prompt "ele sonhava muito", for the syllable based-decoding method. Optional silence is allowed at every node. Considering ele2:lə/0.01 - ele2 is the second syllable of the word "ele", with pronunciation [lə] and negative log-probability of 0.01).

Optional silence is allowed at every node. Sequences of words can also be repeated, as there are continuous back-transitions for full words. Although not shown in the example, if multiple pronunciations are possible for a given word, they are taken into account as alternate pronunciations of a syllable. After decoding, a reconstruction step is needed to join adjacent syllables into their corresponding word, repetition or false start.

The output of both approaches is a per-utterance segmentation into word-relevant segments, be they false starts, repetitions or word-candidates, to be classified in the next stage as mispronounced or not.

B. *Mispronunciation Classification*

The analyzed reading material is extensive, including challenging words and pseudowords. Therefore, the proposed approach targets the possibilities of mispronunciation in a general way, in contrast to considering typical pronunciation errors during decoding. Mispronunciations by children can range from a simple change in one phoneme, or changes in phoneme order, or phoneme deletion or insertion, to severe changes from the intended correct reading. Intonation problems are currently not targeted by our work.

A straightforward possibility to decide if a word pronunciation is correct or not is to compare the uttered sequence of phonemes to the allowable pronunciations. If there is a match, the pronunciation would be considered correct. However, the accuracy of automatic phoneme recognition is not high enough to support this method, since recognition inaccuracies (insertions, deletions and substitutions) can lead to numerous false alarms of mispronunciation. Therefore, methods based on the word likelihood given the correct pronunciation prove to be more successful. We will still apply phoneme recognition to obtain additional inputs for mispronunciation classification. In fact, we will classify word pronunciations based on multiple individual features and combining them in multi-feature classifiers.

For all features that need to consider the reference pronunciation of a word, we allow multiple acceptable pronunciations (based on commonly used pronunciation variants) as well as coarticulation rules depending on neighboring words (where not surrounded by silence). For this task, a neural network based on long temporal context [24] was trained, as mentioned in III.A.1. It is targeted for phoneme recognition and achieved 27% phone error rate on the test set using a free phone-loop model. Its output, used here as the basis of likelihood computations, are state-level posterior probabilities of 34 phones with 3 states each, including non-speech.

1) *Individual features*

Goodness of pronunciation (GOP) [25], [26] is a common metric to detect phonetic mispronunciations by computing the likelihood of a phone realization to belong to the ideal phone that should have been pronounced. We compute GOP-like features on phone posterior probabilities, edit distances of recognized versus ideal phone sequences and other details about the word. Starting with the segmentation of the previous

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2018.2820429, IEEE/ACM Transactions on Audio, Speech, and Language Processing

stage, the most relevant computed features are listed in Table II, with brief descriptions that are expanded in the text. Fig. 5 shows an example of an aligned segment for a word, forced alignment of phonemes, recognized phonemes from a bigram model, and how LLR-spotter and LLR-ali are obtained.

TABLE II. MAIN FEATURES CONSIDERED FOR MISPRONUNCIATION CLASSIFICATION

Feature abbreviation	Summary
LLR-spotter	Log-likelihood ratio based on word-spotting
LLR-ali	Log-likelihood ratio strictly over the original alignment
min-GOP	Minimum (worst) likelihood of a phone from a forced alignment
mean-GOP	Average likelihood of phones from a forced alignment
maxBadPhnProb	Maximum post probability of mismatched (bad) recognized phones
accBadPhnProb	Accumulated post probability of mismatched recognized phones
LevBigram1	Levenshtein edit distance using a bigram phonetic model
LevPL1	Levenshtein edit distance using a constrained phonetic lattice

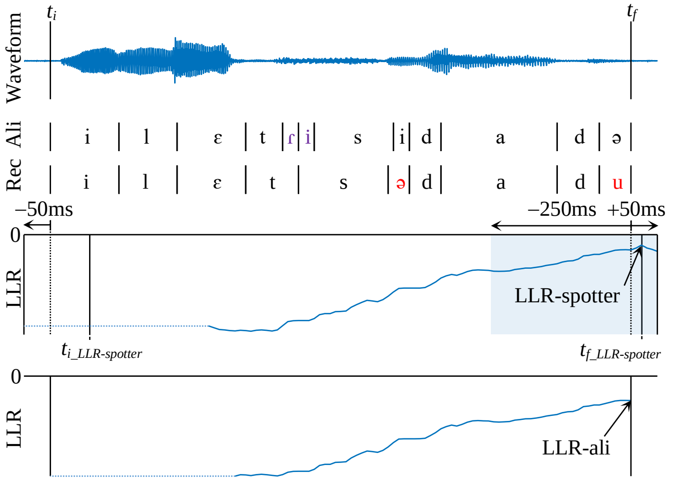

Fig. 5. Schematic of an automatic alignment for the word *eletricidade* [ilɛtɾisidadə], with (top to bottom): waveform signal; forced alignment of reference pronunciation (Ali), recognized phones from a bigram model (Rec) resulting in a Levenshtein distance of 4 (2 deletions and 2 substitutions), LLR from a spotting approach (flexible beginning and end) and LLR with fixed beginning and end.

LLR-spotter: A GOP-like accumulated log likelihood ratio (LLR) from a word spotting approach (LLR-spotter or LLR-s for short). Although the previous stage provides alignments for candidate segments, these may not have the ideal boundaries to calculate likelihood, due to segmentation errors. This can even be the case if segmentation from manual transcription is used (e.g., including some silence inside marked boundaries). LLR-spotter is extracted from a word-spotting approach, as presented in Fig. 6 [27], in the near vicinity of the alignment boundaries (-50ms and +50ms). The keyword model is the sequence of ideal phones and the filler model is the free phone loop. Peak LLR between the models of ideal word and free phone loop is found in the vicinity of the ending time of alignment (-250ms and +50ms, empirically tuned), as shown in Fig.5. The best starting time is in the output token of the keyword model at each frame, as a result of the token-passing decoding approach [28]. The keyword model may also win at different starting times, and new boundary information is obtained. In essence, if only this feature were used, the purpose of the initial segmentation stage would be only to obtain approximate boundaries for candidates.

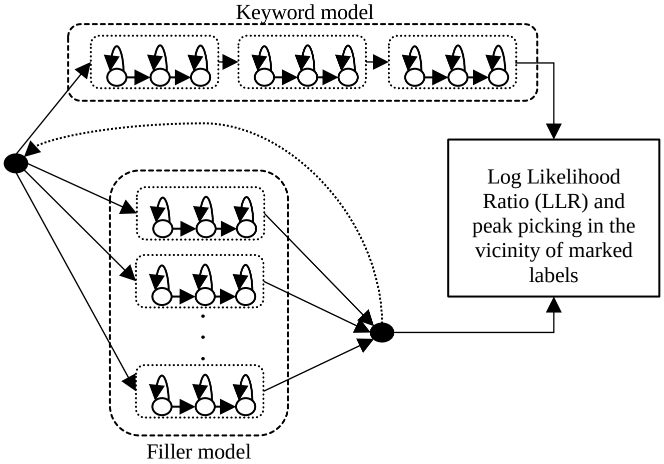
Keyword model
Log Likelihood Ratio (LLR) and peak picking in the vicinity of marked labels
Filler model

Fig. 6. Schematic of obtaining a log-likelihood ratio from a word-spotting approach, using the keyword model of ideal pronunciation versus a filler free phone-loop model [27].

LLR-ali: A log likelihood ratio based strictly on the original segment boundaries. LLR-ali is obtained similarly to the word spotting method, but the starting time has to be the initial frame and the ending at the final frame of the original segmentation as shown in Fig. 5. Although it is considered mainly to compare to LLR-spotter, it might have alternative information for multi-feature classifiers.

min-GOP and mean-GOP: Minimum and average GOP of phones, measured using a posteriori probabilities of phones from the phonetic recognizer neural network. For a forced alignment of ideal phones over the new interval from the word spotting method, the minimum (worst) likelihood of the aligned phones is obtained as a feature, as well as the average likelihood over all phones. We expect that low likelihoods for reference pronunciation phones will indicate mispronunciation.

maxBadPhnProb and accBadPhnProb: Maximum probability and accumulated probability of mismatched phones. As an approximately inverse idea to min-GOP, a free phone loop is used over the posterior probabilities to recognize the uttered sequence of phones. For each recognized phone that does not match the ideal pronunciation, the average posterior probability is taken over its alignment. Both the maximum and sum of these values are taken as features. It is hoped that a mismatched phone with very high probability from the phonetic recognizer will indicate an increased confidence that the word is mispronounced.

Features from phonetic recognition: Levenshtein distances are computed between the ideal phone sequence and the output of two phonetic recognition approaches, for each candidate segment:

- Bigram model. With improved recognition results over a free phone-loop model, a phonetic bigram language model is obtained from the training set and used to decode the best recognized sequence of phones over the candidate segment.

- Phonetic lattice (PL) based on ideal pronunciation. To overcome some errors by the recognizer's output, constrained decoding models are built for each word, based on the notion that the ideal sequence of phones is the most probable to be detected on the segment. Loosely based on an implementation of a bigram model, a less probable back-off with a free phone loop is allowed in addition. The ideal sequence of phones has a higher probability, and only where deviations to this sequence are highly likely does the decoder choose the path of additional phones. An example of the phonetic lattice built for the word *azul* [ɐzul] (blue) is shown in Fig. 7. The path through nodes 0-1-2-3-4-5-7 represents the correct word and those transitions are highly probable (low value of negative log probability). PHN represents all possible phones concurrently (self-loop at node 6), including silence. At the start or after a certain phone of the word, the most probable is the next correct one, as can be expected from a simple bigram model. Although posterior probabilities could also be obtained from the decoded lattices, they are often close to 1 due to the constrained decoding.

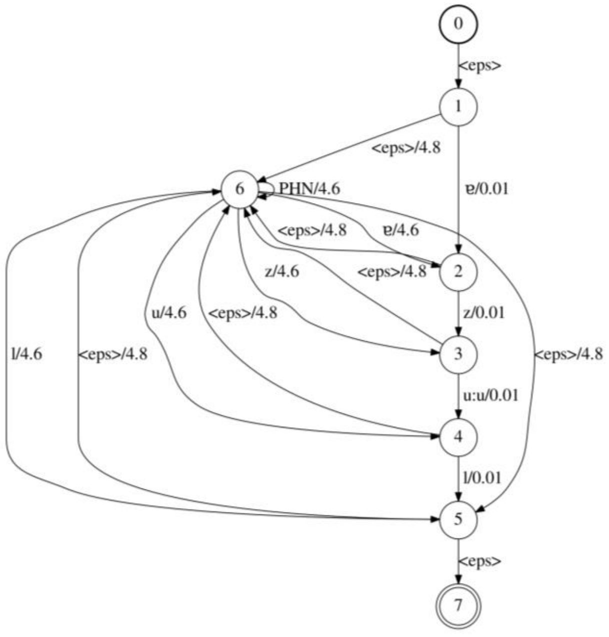

Fig. 7. Example of the constrained phonetic lattice (PL) built for the word azul [ɐzul] (blue).

To compute the Levenshtein distance, phonetic substitutions, deletions and insertions are considered with the same unitary weight (cost). For example, ideal [ɐzul] versus recognized [ɐsul] results in a Levenshtein distance of 1, whereas [zuiS] results in a distance of 3. For each recognition, this distance is taken as a feature (LevBigram1 and LevPL1). Additionally, two slightly different distances are calculated: an edit distance with lower weights for substitutions among similar phonetic groups (LevBigram2 and LevPL2) where, for example, a substitution of [f] for [v] has a lower weight; an edit distance where the substitution weights are based on the phonetic confusion matrix from the output of the phonetic recognizer on the training set (LevBigram3 and LevPL3). For example, if the recognizer often detects [ɔ] for reference [o], this substitution will have lower cost.

Since it is expected that these two phonetic decoding approaches will have differing outputs, additional features are defined by combining the two edit distances, either by the sum or product of the values, for the three types of distances (LevSum1, LevSum2, LevSum3, LevProd1, LevProd2 and LevProd3).

Metrics from LLR-spotter: Based on the best spotted segment obtained in LLR-spotter, the detected number of frames (Nframes) and the LLR area (Area) are also included as features. Area is mostly used for normalizing LLR and is computed by summing the difference of LLR to the best LLR, frame by frame from the beginning to end of the best spotted segment [27].

Difficulty and OLD20: Metrics of word difficulty. It is expected that harder or unfamiliar words are more likely to be mispronounced. The difficulty of the word based on dubious and harder pronunciation rules [22] is considered with and without accounting for word length (Diff1 and Diff2). Additionally, the OLD20 metric of the word is considered, which is the mean Levenshtein distance of the word to its 20 closest orthographic neighbors from a large lexicon [29], which may indicate a degree of familiarity.

Word length: Additional features include the number of phones of the closest allowable pronunciation (Nphones), its number of graphemes (Ngraph) and number of syllables (Nsylls).

LLR normalizations and interactions: Several normalizations and interactions of LLR-spotter with other features are considered, by division or product, represented as, e.g., LLR-s/Nframes or LLR-s*LevBigram1.

2) *Multi-feature models*

Our goal is to classify whether a word is mispronounced or not, with mispronunciation being the positive class. Therefore, we consider the task a problem of binary classification. If only one feature is analyzed, a decision threshold can simply be defined for a hard decision (yes or no). However, the optimal operating points may vary and it is preferable to analyze the performance of selecting several thresholds, usually with detecting error trade-off (DET) curves. Toward that end, multi-feature models that can output continuous values are preferred. Although continuous outputs could be interpreted as degrees of correctness of pronunciations, we do not explore this interpretation here.

To combine the information of several features, aiming to improve the classification of mispronunciations, we investigate the following models taking multiple inputs:

- Logistic regression (Logit). A logistic regression model for a binomial distribution, a case of generalized linear regression, is trained through maximum likelihood estimation. The logistic function (1) gives response probabilities by the linear combination of predictive features.

$$\hat{y} = \frac{1}{1+e^{\mathbf{a}^T\mathbf{x}+b}} \quad (1)$$

In (1), $\hat{y}$ is the predicted output, ranging between 0 and 1, corresponding to the probability of the sample being in the mispronounced class based on a linear combination of features where $\mathbf{x}$ is the feature vector, $\mathbf{a}$ is the coefficient vector (weights) of the input features and b is the intercept (bias) term.

- Neural networks (NNs). Networks are built with one hidden layer with variable number of neurons and one output, trained with Levenberg-Marquadt backpropagation [30] and optimizing cross-entropy. The transfer function for the hidden neurons is the hyperbolic tangent sigmoid and at the output layer a logistic sigmoid function is used, providing output between 0 and 1.
- Support vector machines (SVMs). SVMs for binary classification are trained with a second order polynomial kernel, *C* parameter of 0.1 and an automatic heuristic kernel scale. To obtain continuous values, the considered output is not the binary decision but the classification score, which is the distance of the input vector to the decision boundary.

The hyperparameters for NNs and SVMs were chosen empirically. To avoid over-fitting to the training set, an alternative to using the entire set of features is analyzed. Stepwise feature selection is applied [31], through two approaches: adding features step by step when no features are included (Step-add) and removing features step by step when all features are included (Step-remove). For Step-add, we select the feature that minimizes deviance[2] when a logistic regression is applied. However, a feature is only added if the decrease in deviance is statistically significant according to a chi-square test ($p < 0.05$). Similarly, for Step-remove, features are removed if, with their presence, the increase in deviance has a $p > 0.10$. This usually leads to different features being selected by the two approaches, with a logistic regression applied at the end. NN and SVM are again analyzed by using only the selected features as input (NN-step and SVM-step).

IV. Results and Discussion

A. Segmentation

Using both the word-based decoding with silence cutting and the syllable-based decoding for the segmentation stage, the overall word error rate (WER) and the detection of extra speech events (repetitions and false starts) can be analyzed.

WER is analyzed by comparing the decoded sequence of words and events to the reference transcription. Using the full text of the original prompts as hypothesis, WER is 9%, where errors correspond to repetitions, false starts, insertions and deletions. Since the decoding strategies do not take insertions and deletions into account, these will always appear as errors. WER values for the test set, when using the optimal insertion penalty for the training set, for the baseline without cutting silence and both segmentation approaches are presented in Table III, as well as results when using the optimal penalty for the test set.

Table III. Overall word error rate (WER) and miss and false alarm rates for the detection of repetitions and false-starts, for the test set, using the optimal insertion penalty of the train set and the best one on the test set (best).

Segmentation approach	WER %	Miss %	False Alarm %	WER % (best)	Miss % (best)	False Alarm % (best)
Baseline	4.15	15.92	2.36	4.11	17.60	2.21
Word-based w/ sil. cut	2.45	11.17	0.98	2.41	10.61	0.98
Syllable-based	2.17	11.45	0.68	2.16	10.89	0.70

We consider recognition of repetitions and false starts as a detection task, lumping both into a single class. Although the false starts allowed are up to the last syllable, in the transcribed data some are complete mispronunciations of a word, with a subsequent attempt of correction. Those are possibly detected as repetitions with these methods and motivate analyzing the detection of both repetition and false start events as one class. To evaluate a system's performance in the detection of these events, we stipulate that:

- Extra events (insertions) are false alarms;
- Undetected events (deletions) are misses;
- Events detected as belonging to a different word (substitution) are also misses. For example, a false start of one word may be detected as a repetition of the previous one.

These specifications are similar to the ones used in NIST evaluations [33]. However, to calculate false alarm rates, the number of false alarms are divided by the total number of original words. Fig. 8 shows the detection error trade-off (DET) for the segmentation approaches on the test set, obtained by using a wide search beam during decoding and various word insertion penalties and lattice rescoring weights. Operating points correspond to using insertion penalties and lattice weights that resulted in the best WER on the training set.

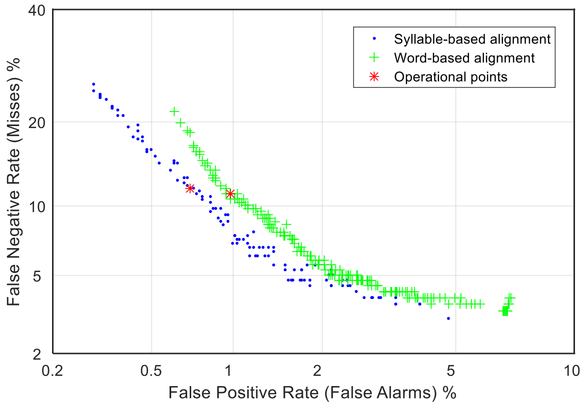

Fig. 8. Detection error tradeoff (DET) for the detection of repetitions and false starts on the test set, for both decoding approaches. Operating points are with the best train insertion penalties.

[2] Deviance can be computed by the sum of unit deviances given for a binomial distribution by $2\{y\log(y/\mu)+(1-y)\log((1-y)/(1-\mu))\}$ with observation y and prediction μ [32].

The equal error rate (EER) for the systems is not of interest since it corresponds to relatively high false-alarm rate (equal to the miss error rate) and higher WER, far from the targeted operating points. Table III presents the resulting values at these points, as well as the best possible ones by optimization on the test set.

Comparing the WER results as well as the DET values, it is clear that the second approach – syllable-based decoding without cutting silent segments – performs better than the alternative method of cutting silent segments and aligning full words. Still, the above results do not take into account the time-wise alignment information. Comparing the decoded hypotheses to the manual transcription (reference), three metrics for alignment match are analyzed and presented in Table IV:

- overlapRef – percentage of frames that overlap per event over the length of the reference word, averaged for all events.
- overlapOverMax – percentage of frames that overlap per event over the maximum interval between the reference and hypothesis start and end frames, averaged over all events. This metric penalizes longer hypotheses.
- overlapUtt – percentage of overlap frames per utterance over the length of all events of the sentence, averaged over all utterances.

For each event of the reference, only one hypothesis is compared: the one that corresponds to the same word and with the largest overlap if more than one exists. Therefore, all the metrics penalize shorter hypotheses.

TABLE IV. METRICS ANALYZING FRAME-WISE ALIGNMENT MATCH BETWEEN THE MANUAL TRANSCRIPTION AND DECODED HYPOTHESES.

Segmentation approach	**overlapRef %**	**overlapOverMax %**	**overlapUtt %**
Baseline	89.00	82.52	90.26
Word-based w/ sil. cut	89.71	84.42	91.98
Syllable-based	90.21	83.71	92.68

The alignment metrics also show that the syllable-based decoding performed better overall, although a smaller overlapOverMax shows that it may have provided slightly larger segments. Subsequently, only this automatic segmentation method will be used to analyze mispronunciation classification results. The improved alignment accuracy justifies the added complexity and computation in decoding. It also has the advantage of skipping a non-speech removal step that needs to decide a minimum duration for non-speech segments to be cut. Leaving this decision as optional silence in the decoder seems to be ideal.

B. Mispronunciation classification

Given candidate word segments with information about start and end times and corresponding prompt word label, an automatic classifier decides whether the word was mispronounced (positive class, 1) or not (negative class, 0). Using continuous values for predictions or probabilities of belonging to a class, we can use each output value as a threshold for decision and derive DET curves.

Two sources of candidate segments will be analyzed: manual segmentation from the manual transcription and automatic segmentation from the syllable-based automatic decoding. We expect that manual segmentations provide the best results and the clearest analysis of which features are important to classify mispronunciations. With automatic segmentation, we expect some misalignments with the ground truth (manual segmentation). However, there must be an overlap of alignment in order for them to be considered a match.

Two groupings for mispronounced classes will be analyzed:

- SUB+PHO: all mispronunciations as the positive class;
- SUB: only severe mispronunciations (SUB) as the positive class, without considering slight mispronunciations (PHO), since the latter are usually harder to detect.

To compare classifications, given continuous output values for candidates, we will be analyzing false alarm and miss rates. Since positive samples (mispronunciations) occur much less frequently than negative ones (correct words), the maximum accuracy measure would often relate to very low false alarm rates but with miss rates higher than 50%. Other measures such as F-score do not take into account the number of true positives. To target more interesting operating points, we found it best to combine false alarm rate (FA) and miss rate in a weighted cost metric (2), where minimal cost is better.

$$Cost = w_1 \cdot FA + w_2 \cdot Miss \tag{2}$$

In (2), w_1 is the weight given to false alarms and w_2 is the weight of misses. This way, more weight can be given to false alarms, moving toward fewer false alarms than an equal error rate, but not so much as to reach miss rates higher than, for example, 50%. We decided to target optimal cost around 5% false alarm, with w_1 defined as 1 and w_2 defined as 0.33.

The point of minimum cost in the training set will define the decision threshold. For all the considered groupings (manual and automatic segments, SUB and SUB+PHO as classes), two separate analyses of classifier model training and testing will be done:

- Cross-validation over the training set (CV-train). A 5-fold cross-validation is done using the training data used for acoustic models. The results are obtained by aggregating the outputs on the test data in each fold.
- Test. Predictions on the test set are made by training a model over the entire training set.

For models that depend on random initialization (NN weights and SVM automatic heuristic kernel scale), results will be of the model of minimum cost over 10 runs on training data.

Table V summarizes the results of the obtained cost metric when using only the individual features for classification of the SUB+PHO class. Features that are not shown, such as difficulty and number of phones, provide comparatively very poor results individually. Further normalizations and interactions provide either similar or slightly worse results compared to the displayed ones. LLR-spotter proves to be the best performing feature, with a significant improvement over the similar LLR-ali metric, where the initial alignment is used, even for the manual segmentation.

TABLE V. COST RESULTS FOR THE CLASSIFICATION OF SUB+PHO CLASS VS. CORRECT WORDS, USING INDIVIDUAL FEATURES.

	CV-train		**Test**	
Feature	**Manual**	**Auto**	**Manual**	**Auto**
LLR-spotter	**0.136**	**0.141**	**0.157**	**0.163**
LLR-ali	0.171	0.191	0.187	0.192
min-GOP	**0.194**	**0.200**	**0.234**	**0.233**
mean-GOP	0.246	0.251	0.276	0.273
maxBadPhnProb	0.280	0.248	0.288	0.276
sumBadPhnProb	**0.232**	**0.231**	**0.241**	**0.247**
LevBigram1	0.248	0.247	0.263	0.262
LevPL1	0.227	0.235	0.242	0.252
LevSum1	**0.199**	**0.206**	**0.212**	**0.221**
LLR-s/Nphones	0.156	0.168	0.187	0.190
LLR-s*LevBigram1	0.159	0.162	0.179	0.185
LLR-s*LevPL1	0.179	0.194	0.203	0.208

LevPL1, the Levenshtein distance by using the constrained phonetic lattices (PL), provides a better cost metric than the bigram one (LevBigram1), with their combination proving successful (LevSum1). Analyzing the phone error rate (PER) for the two phone decoding systems over candidate segments (manual transcription), for correct words and for mispronunciations, as shown in Table VI, provides an interesting insight. As expected, the constrained phonetic lattice results in a low PER for correct words, since the sequence of correct pronunciation is much more probable. On the other hand, for mispronunciations, the PER is higher using phone lattices since it has less freedom to recognize mispronounced phones. For the bigram, the higher PER on mispronunciations than correct words may reflect some problems of the manual transcription, as it is often hard to decide which sequence of phones was uttered in mispronunciations.

TABLE VI. PHONE ERROR RATES (PER) OF THE TWO PHONE DECODING APPROACHES.

Phone decoding model	**PER %** **correct word**	**PER %** **mispronunciations**	**PER %** **all**
Bigram	22.81	36.03	24.12
Phonetic lattice	2.57	41.46	6.41

Returning to classification results, we can show the effect of selecting multiple thresholds of Levenshtein distance to classify candidates as mispronounced or not by plotting the DET of LevBigram1, LevPL1 and LevSum1 (their sum), as in Fig. 9. As can be seen, PL performs worse at finding all mispronunciations, never going below 43% miss rate (with a threshold of distance 1). However, PL seems better for lower false alarms and the combination of both features clearly provides improved results.

Comparing the use of manual segmentation versus automatic segmentation, the automatic one does result in slightly worse, albeit close, results, as shown in Fig. 10. For a 5% false alarm rate, a 33.51% miss rate is obtained by LLR-spotter from manual segmentation, versus a 35.32% miss rate using automatic segmentation.

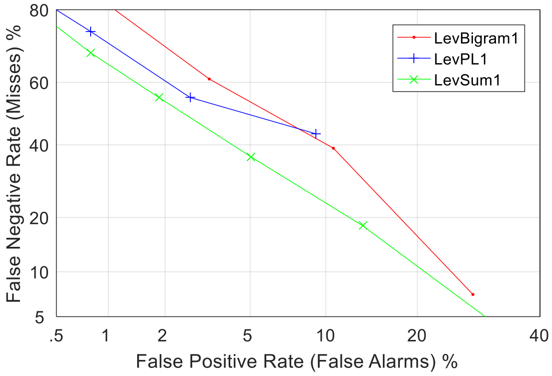

Fig. 9. DET of Levenshtein edit distance features, for the classification of the SUB class, on the test set.

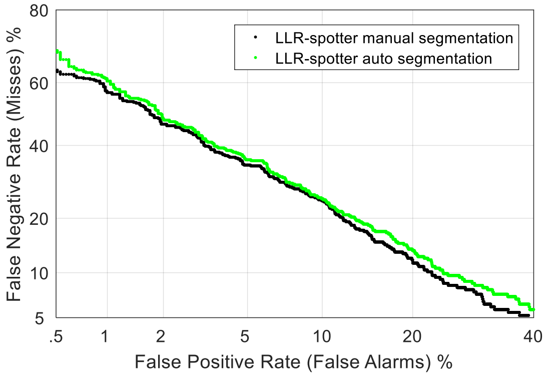

Fig. 10. DET of LLR-spotter for the SUB+PHO classification on the test set by using manual or auto segmentations.

Table VII summarizes the results of using the multi-feature models described in section 3.2.2, with the goal of combining the information of several features to improve classification. NN-step and SVM-step represent the use of the selected features by the best feature selection method for the same conditions (either Step-add or Step-remove). In addition to the cost obtained by the optimal thresholds from training, miss rates for the same 5% false alarm rate are indicated, although the operating points for the given costs vary slightly from 4% to 6% false alarm rate.

TABLE VII. COST AND MISS RATES AT A 5% FALSE ALARM FOR THE CLASSIFICATION OF SUB+PHO CLASS VS. CORRECT WORDS, USING MULTI-FEATURE MODELS (LLR-SPOTTER INCLUDED FOR COMPARISON).

	CV-train				**Test**			
	Manual		**Auto**		**Manual**		**Auto**	
Classification Model	**Cost**	**Miss**	**Cost**	**Miss**	**Cost**	**Miss**	**Cost**	**Miss**
LLR-spotter	0.136	27.17	0.141	29.23	0.157	34.03	0.163	35.58
Logit-all	0.121	22.67	0.137	27.34	0.137	29.61	0.154	33.51
Step-add	0.120	22.35	0.136	27.26	0.141	29.61	0.150	33.25
Step-remove	0.121	22.59	0.136	27.22	0.140	29.87	0.153	33.51
NN	0.119	21.87	0.132	25.89	**0.136**	**27.79**	0.144	30.13
NN-step	**0.116**	**21.35**	0.133	26.17	0.140	28.83	0.144	32.47
SVM	0.117	22.03	**0.130**	**25.29**	0.139	29.35	**0.142**	**29.35**
SVM-step	0.118	21.63	0.130	25.53	0.139	30.65	0.152	33.77

A significant improvement was obtained by considering multiple features, and the best classifiers vary from neural networks to SVMs, with similar results. For the manual segmentation, neural networks provided better results, and using feature selection was greatly helpful for CV-train. For the other cases, stepwise feature selection was not helpful and the best results for automatic segmentation were obtained with SVMs. For the same 5% false alarm rates, the improvement of miss rate relative to the best individual feature (LLR-spotter) are 22% and 13% in CV-train (manual: 27.17% to 21.35%; auto: 29.23% to 25.29%) and 18% on the test set (manual: 34.03% to 27.79 %; auto: 35.58% to 29.35%).

Even if stepwise feature selection was only helpful for manual segmentation, analyzing which features are consistently selected may give an insight into the most relevant ones. For the Step-add feature selection, these are the features that are consistently selected for all the folds of cross-validation on the training set, for SUB+PHO:

- LLR-spotter;
- LLR-ali;
- mean-GOP;
- maxBadPhoneProb;
- LevDistPL1 or LevDistPL3;
- 1 combination of Levensthein distances - either LevSum3 or LevProd3;
- 1 normalization of LLR-spotter - LLR-s/Nchars, LLR-s/Nframes or LLR-s/Area;
- 1 interaction of LLR-spotter with phone lattice distance - LLR-s*LevPL1 or LLR-s*LevPL3.
- Area (LLR area from the spotting approach).

Most of the designed features prove to be relevant and apparently carry complementary information that enhances mispronunciation classification. Curiously, LLR-ali was always selected, even though it performs worse than LLR-spotter. It may be useful for cases where something extra is said at the beginning or end of words (for example, adding a plural suffix) and where by using the spotting approach on these segments, a correct pronunciation is found (LLR-spotter would hurt the classification), whereas the original segmentation encompassed the mispronunciation. Furthermore, mean-GOP and maxBadPhoneProb were chosen over min-GOP and sumBadPhoneProb. Effectively, even if min-GOP and sumBadPhoneProb are better individually (in minimum cost and in the stepwise criterion of deviance), after the first step when LLR-spotter is added to the stepwise model, mean-GOP and maxBadPhoneProb would provide better results if added. This is due to these two selected features having less correlation or sharing less information with LLR-spotter and other important features, and helping the model with alternative information.

To further analyze the improvement of using multi-feature models over LLR-spotter, Fig. 11 shows the DET curves comparing the use of the individual LLR-spotter feature versus a multi-feature model, in this case, the neural network using all features.

Overall, be it only LLR-spotter or a multi-feature model, we observe from the edges of the DET curves that there are cases of mispronunciation that are hardly detected, only with very high false alarm rates. There are also cases of correct word pronunciations that easily result in false alarms. Most of these, where the manual annotator did not indicate mispronunciation, were found to be due to two factors: noise simultaneous with speech and words with low vocal effort (whispering). Words with a low vocal effort often occur at the end of sentence with the final syllables of the word appearing unvoiced. We attempted to add as features the word position in the sentence and a binary feature for being the last word, but they were never helpful.

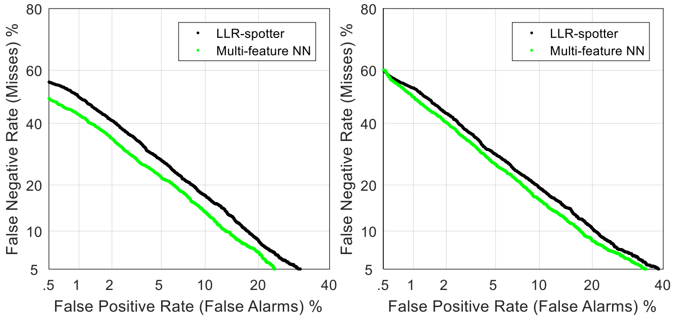

Fig. 11. DET curves of LLR-spotter and multi-feature neural network using manual (left) and automatic (right) segmentations, for SUB+PHO CV-train classification.

There are two further main problems to tackle. The first is that the output of the phonetic recognizer is prone to errors, otherwise the match of the recognized phones to reference pronunciation would suffice. This was addressed by including several features that compensate for misrecognitions in some fashion (e.g., probability of mismatched phones and Levenshtein distance from a constrained phone decoding where the ideal sequence is highly probable). Nevertheless, by improving the accuracy of the phonetic recognizer, better results can be expected. The second problem is the subjective manual annotation of correct words and mispronunciations, where many cases are dubious for different manual annotators. Fixing annotator errors could have an effect on results but the methodology itself might not change. Not much can be done from an automatic perspective other than improving the reference by combining the opinions of multiple annotators.

V. Conclusion

We have proposed a system for automatically detecting common mispronunciations and disfluencies in children's reading. Automatic processing in two steps – segmentation to obtain candidate word pronunciation segments, and classification as mispronounced or not – provides small differences compared to manual transcriptions. We address the common problem of intra-word pauses with a syllable-based decoding, giving better segmentations than our previous methods. For mispronunciation detection, combining several features that may have alternative information improved results significantly, compared to using only one log-likelihood ratio metric. Combining the output of two phone recognition models (a bigram and a constrained phonetic lattice) provided more information about mispronunciation than one strategy alone.

We note that some aspects of our work may be optimized for the conditions of the data being analyzed. For example, not allowing deletions of words during segmentation worked better than allowing them in the alignment lattice, but it may not be so for less controlled scenarios with more unprompted and incomplete utterances. Still, there are other clear enhancements to pursue: dealing with utterances with low vocal effort and improving phonetic recognition models. The segmentation stage might also be improved to account for the need to align ideal pronunciations to incorrect attempts, possibly by allowing a garbage model to match mispronounced phones or unrelated extra segments. Speaker adaptation might also be useful at all stages, including adjusting phonetic recognition to individual reading speeds.

It has been shown in previous work [10], [11] that using rates of disfluencies in addition to reading speed metrics can improve the prediction of a child's reading performance, and the impact of the improvements given by the methodology proposed in the current work needs to be investigated. Applying the proposed methods to children older than 10 years would probably mean that new acoustic models would have to be built due to severe changes in the children's voices. Additionally, the frequency of disfluent events (such as intra-word pauses) could be different, which would mean that the probabilities in models should be adjusted and that the features relevant for classification may even change.

VI. Acknowledgement

The authors acknowledge the support of Microsoft to this project.

VII. References

[1] J. Lopes, L. Spear-Swerling, C. Oliveira, M. Velasquez, L. Almeida, and L. Araújo, *Ensino da leitura no 1^o ciclo do ensino básico*. Fundação Francisco Manuel dos Santos, 2014.

[2] National Reading Panel, "Teaching children to read: An evidence-based assessment of the scientific research literature on reading and its implications for reading instruction," National Institute of Child Health and Human Development, USA, 2000.

[3] L. S. Fuchs, D. Fuchs, M. K. Hosp, and J. R. Jenkins, "Oral Reading Fluency as an Indicator of Reading Competence: A Theoretical, Empirical, and Historical Analysis," *Scientific Studies of Reading*, vol. 5, no. 3, pp. 239–56, 2001.

[4] J. Mostow, S. F. Roth, A. G. Hauptmann, and M. Kane, "A Prototype Reading Coach That Listens," in *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, Menlo Park, CA, USA, 1994, pp. 785–792.

[5] M. Black, J. Tepperman, S. Lee, P. Price, and S. Narayanan, "Automatic detection and classification of disfluent reading miscues in young children's speech for the purpose of assessment," in *Proc. Interspeech 2007*, Antwerp, Belgium, 2007, pp. 206–209.

[6] J. Duchateau *et al.*, "Developing a reading tutor: Design and evaluation of dedicated speech recognition and synthesis modules," *Speech Communication*, vol. 51, no. 10, pp. 985–994, Oct. 2009.

[7] D. Bolaños, R. A. Cole, W. Ward, E. Borts, and E. Svirsky, "FLORA: Fluent Oral Reading Assessment of Children's Speech," *ACM Trans. Speech Lang. Process.*, vol. 7, no. 4, p. 16:1–16:19, Aug. 2011.

[8] M. P. Black, J. Tepperman, and S. S. Narayanan, "Automatic Prediction of Children's Reading Ability for High-Level Literacy Assessment," *Trans. Audio, Speech and Lang. Proc.*, vol. 19, no. 4, pp. 1015–1028, May 2011.

[9] J. Duchateau, L. Cleuren, H. V. hamme, and P. Ghesquière, "Automatic assessment of children's reading level," in *Proc. Interspeech 2007*, Antwerp, Belgium, 2007, pp. 1210–1213.

[10] J. Proença, C. Lopes, M. Tjalve, A. Stolcke, S. Candeias, and F. Perdigão, "Automatic Evaluation of Children Reading Aloud on Sentences and Pseudowords," in *Proc. Interspeech 2017*, Stockholm, Sweden, 2017.

[11] J. Proença, C. Lopes, M. Tjalve, A. Stolcke, S. Candeias, and F. Perdigão, "Automatic evaluation of reading aloud performance in children," *Speech Communication*, vol. 94, pp. 1–14, Nov. 2017.

[12] J. Proença, D. Celorico, C. Lopes, S. Candeias, and F. Perdigão, "Automatic Annotation of Disfluent Speech in Children's Reading Tasks," in *Proc. IX Jornadas en Tecnologías del Habla and V Iberian SLTech Workshop - IberSPEECH'2016*, Lisbon, Portugal, 2016, pp. 172–181.

[13] S. M. Abdou *et al.*, "Computer aided pronunciation learning system using speech recognition techniques," in *Proc. Interspeech 2006*, Pittsburgh, USA, 2006, pp. 849–852.

[14] T. Cincarek, R. Gruhn, C. Hacker, E. Nöth, and S. Nakamura, "Automatic pronunciation scoring of words and sentences independent from the non-native's first language," *Computer Speech & Language*, vol. 23, no. 1, pp. 65–88, Jan. 2009.

[15] Y. Liu, E. Shriberg, A. Stolcke, and M. P. Harper, "Comparing HMM, maximum entropy, and conditional random fields for disfluency detection," in *Proc. Interspeech*, 2005, pp. 3313–3316.

[16] H. Medeiros, H. Moniz, F. Batista, I. Trancoso, L. Nunes, and others, "Disfluency detection based on prosodic features for university lectures.," in *Proc. Interspeech*, Lyon, France, 2013, pp. 2629–2633.

[17] H. Moniz, F. Batista, A. I. Mata, and I. Trancoso, "Speaking style effects in the production of disfluencies," *Speech Communication*, vol. 65, pp. 20–35, Nov. 2014.

[18] E. Yilmaz, J. Pelemans, and H. V. hamme, "Automatic assessment of children's reading with the FLaVoR decoding using a phone confusion model," in *Proc. Interspeech 2014*, Singapore, 2014, pp. 969–972.

[19] A. Hagen, B. Pellom, and R. Cole, "Highly Accurate Children's Speech Recognition for Interactive Reading Tutors Using Subword Units," *Speech Communication*, vol. 49, no. 12, pp. 861–873, Dec. 2007.

[20] X. Li, Y.-C. Ju, L. Deng, and A. Acero, "Efficient and Robust Language Modeling in an Automatic Children's Reading Tutor System," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, vol. 4, pp. 193–196.

[21] J. Proença, C. Lopes, M. Tjalve, A. Stolcke, S. Candeias, and F. Perdigão, "Detection of Mispronunciations and Disfluencies in Children Reading Aloud," in *Proc. Interspeech 2017*, Stockholm, Sweden, 2017.

[22] J. Proenca, D. Celorico, S. Candeias, C. Lopes, and F. Perdigão, "The LetsRead Corpus of Portuguese Children Reading Aloud for Performance Evaluation," in *Proc of the 10th edition of the Language Resources and Evaluation Conference (LREC 2016)*, Portorož, Slovenia, 2016.

[23] D. Povey *et al.*, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, Hilton Waikoloa Village, Big Island, Hawaii, US, 2011.

[24] FIT, "Phoneme recognizer based on long temporal context, Brno University of Technology," 06-May-2015. [Online]. Available: http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context. [Accessed: 06-May-2015].

[25] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. 2–3, pp. 95–108, Feb. 2000.

[26] T. Pellegrini, L. Fontan, J. Mauclair, J. Farinas, and M. Robert, "The Goodness of Pronunciation algorithm applied to disordered speech," presented at the The 15th Annual Conference of the International Speech Communication Association - INTERSPEECH 2014, 2014, pp. 1463–1467.

[27] A. Veiga, C. Lopes, L. Sá, and F. Perdigão, "Acoustic Similarity Scores for Keyword Spotting," in *Computational Processing of the Portuguese Language*, J. Baptista, N. Mamede, S. Candeias, I. Paraboni, T. A. S. Pardo, and M. das G. V. Nunes, Eds. Springer International Publishing, 2014, pp. 48–58.

[28] S. Young *et al.*, *The HTK book*, vol. 2. Entropic Cambridge Research Laboratory Cambridge, 1997.

[29] T. Yarkoni, D. Balota, and M. Yap, "Moving beyond Coltheart's N: A new measure of orthographic similarity," *Psychonomic Bulletin & Review*, vol. 15, no. 5, pp. 971–979, Oct. 2008.

[30] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[31] N. R. Draper and H. Smith, *Applied regression analysis*, 3rd ed. Wiley, 1998.

[32] P. Song, *Correlated Data Analysis: Modeling, Analytics, and Applications*. New York, NY: Springer New York, 2007.

[33] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, “Results of the 2006 spoken term detection evaluation,” in *Proc. SIGIR 2007*, Amsterdam, Netherlands, 2007, vol. 7, pp. 51–57.