Subword-based Multi-Span Pronunciation Adaptation for Recognizing Accented Speech

Timo Mertens
Department of Electronics and Telecommunications
Norwegian University of Science and Technology, Norway
mertens@iet.ntnu.no

Kit Thambiratnam, Frank Seide
Microsoft Research Asia
Beijing, China
{kit,fseide}@microsoft.com

***Abstract*—We investigate automatic pronunciation adaptation for non-native accented speech by using statistical models trained on multi-span lingustic parse tables to generate candidate mispronunciations for a target language. Compared to traditional phone re-writing rules, parse table modeling captures more context in the form of phone-clusters or syllables, and encodes abstract features such as word-internal position or syllable structure. The proposed approach is attractive because it gives a unified method for combining multiple levels of linguistic information. The reported experiments demonstrate word error rate reductions of up to 7.9% and 3.3% absolute on Italian and German accented English using lexicon adaptation alone, and 12.4% and 11.3% absolute when combined with acoustic adaptation.**

I. Introduction

Robustness to non-native accent variation still remains unsolved for Automatic Speech Recognition (ASR). In large vocabulary continuous speech recognition (LVCSR) acoustic models (AM) and language models (LM) are usually adapted to improve accuracy for a speaker. In this contribution we focus on adapting the pronunciation lexicon for a set of non-native speakers. Inconsistent pronunciations with respect to the native canonical pronunciation of a word are especially frequent in non-native speech, e.g., German-accented English. Non-native inconsistencies are influenced by the linguistic aspects of both the speaker's native language (L1) as well as his non-native language (L2). As LVCSR models, especially the lexicon, are optimized for native speech, a mismatch between models and accented speech leads to a substantial increase in recognition error rates [1].

On the lexicon-side of the adaptation chain, most pronunciation adaptation approaches address the problem by including multiple weighted pronunciations for a word in the lexicon. These pronunciation variants are generated by applying re-writing rules, i.e. rules that change the identity of a phone in a given context, to pronunciations in the lexicon. The task of a pronunciation model is then to learn these rules and assign a probability to them. Generating rules is usually divided into knowledge-based and data-driven techniques (see [2] for an extensive overview). Where the knowledge-based approach utilizes additional sources such as expert linguistic knowledge to propose rules, data-driven techniques use the actual ASR output to learn pronunciation patterns. Speaker adaptation techniques such as maximum likelihood linear regression (MLLR) have also been investigated, where the goal is to directly capture pronunciation variation in the emission states of the AMs. Although fairly successful, AM adaptation has difficulties addressing more complex mispronunciation phenomena such as insertions and deletions. Pronunciation adaptation for non-native accented speech generally follows similar ideas [3].

In this contribution we investigate LVCSR adaptation techniques for recognizing accented English. A significant shortcoming of traditional lexicon adaptation approaches is the limited modeling context: re-writing rules often operate on a phone level while conditioning only on the immediate phone neighbors. For this reason we investigate multiple streams of multi-span subword segmentations to capture varying amounts of context when estimating non-native pronunciation variants from speech. Both, linguistic research as well as our own analyses have shown that mispronunciation patterns are often realized on a unit larger than the phone [4], which we try to explain with a simple linguistic model in Sec. 2. The core of the adaptation framework is a subword parser which breaks down words of the target language into parse tables which encode the different subword segmentations. The parser is deterministic as the different segmentations only depend on the syllabification of the word which allows for consistent subword segmentations and thus reliable confusion estimates. Phone transcripts of accented training speech are aligned with the different segmentations to train a statistical model to generate and score mispronunciations. With this model a native lexicon can be adapted by interpolating scores from different layers for each hypothesized lexicon pronunciation. In the overall LVCSR adaptation chain we also include traditional AM adaptation which gives additive gains w.r.t. lexicon adaptation.

The paper is organized as follows: Sec. 2 describes a simple linguistic model of accented speech production. Sec. 3 and 4 introduce the training of the model and describe lexicon adaptation. In Sec. 5 and 6 we explain the experimental setup and present evaluation results. Sec. 7 concludes the paper.

II. Model of Accented Speech Processing

In this section we present a simplistic model of human speech perception and production. We utilize this model to motivate and illustrate the use of certain linguistic constraints employed in the adaptation framework described in Sec. 3. Our model draws knowledge from related fields such as second language acquisition and human language learning which most LVCSR adaptation approaches ignore. For example, [5] describes that the main reason for non-native accent is a lack in accuracy when perceiving an L2 sound. Failing to discriminate the L2 sound from native L1 sounds contributes to the inability to learn how to produce the sound correctly, as it will be *assimilated* with a native L1 sound.

In Fig. 1 we argue that when listening, the non-native speech is first segmented into units. The lexical label of the word is then mapped to its unit segmentation in a mental lexicon for later look-up. At the foundation of the model lies the unit

TABLE I
SEGMENTATIONS FOR "THROUGH" WITH A GERMAN UNIT INVENTORY.

Syllable	/th r uw/	*/th r uw/*	/?/
PSC	/th r/ /uw/	*/th r/* /uw/	/?/ /uw/
Phone	/th/ /r/ /uw/	*/th/* /r/ /uw/	/?/ /?/ /uw/

inventory which encodes all possible subword units of the speaker's language. The question now is how these subword units are defined in terms of length (e.g., number of phones) and linguistic attributes (e.g., grammatical tag). We make two strong assumptions at this point: first we assume units to be of segmental nature, i.e. we ignore alternative subword representations such as asynchronous articulatory features as used in [6]. Second we only consider linguistically motived units. The reason for that is strong evidence in linguistic research [4] that pronunciation variation is influenced by the linguistic subword structure of the language.

Take as an example the word /th r uw/ ('through') in Table 1 with three segmentations for its pronunciation: phones, position-specific cluster (PSC) [7] and syllables. The length of the chosen subword unit dictates a tradeoff between modeling context and data sparseness: whole word or syllable units model context extremely well, yet are prone to data sparseness as the underlying inventory is huge. Smaller units, such as phones, are well represented in the data but lack in context.

When a learner of a foreign language perceives L2 speech he segments the speech according to his unit inventory, which, initially, only contains L1 units. Our model explains non-native mispronunciations then as an incorrect segmentation due to so-called out-of-unit (OOU) segments, i.e. segments that are not in the unit inventory of the speaker, e.g., red italic units in columns 3 and 4 in Table 1. As German does not have /th/ nor the English /r/, all units containing these phones will OOU and prone to accent. As the learner gains L2 proficiency, he creates new entries in his unit inventory for OOUs and is thus able to realize L2 units correctly. We call this intermediate level L1.5. Even if a unit is in the inventories of L1 and L2, other linguistic features associated with that unit can cause it to be an OOU in L2. For example, German has /d/ in its inventory, yet when augmented with a feature for word position it can become an OOU as German does not allow voiced plosives word final. Other examples are complex codas in Italian which are often split by inserting an /ax/ such that the closed syllable is forced into two open syllables, e.g., /p aa r t/ → /p aa r . t ax/. This shows the importance of going beyond phones by considering the L1 and L2 subword structures and features.

In summary, we explain non-native mispronunciations as a mismatch between the L1 and L2 unit inventories, which in turn causes incorrect segmentation and accented production of non-native speech. The next section describes how this fundamental assumption is realized when statistically modeling mispronunciations for lexicon adaptation.

III. STATISTICAL ACCENT MODEL

In order to model the theory described in Sec. 2 we need to know the non-native L1.5 unit inventory and the speaker's mapping function. As this information is not given, the task is to learn both from data. If we have a canonical parse table for a word, and a set of accented pronunciations of that same word, we can align both to 1) identify which units in the inventory are subject to variation and 2) the mapping function that determines what they will be mispronounced as. First we describe the

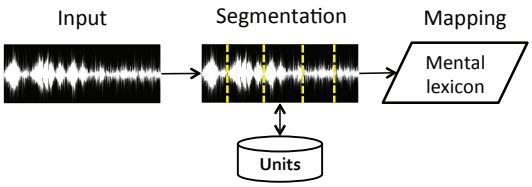

Fig. 1. Simplistic model of speech perception.

model's underlying structure, focusing on how to approximate a speaker's L1.5 unit inventory with varying multi-span subword segmentations. Given some training data, we then show how to train a statistical model that is able to generalize from observed to unseen words. Finally we demonstrate how to predict and score a set of mispronunciations based on the trained model.

A. Subword Parse Tables

In this contribution we consider three subword segmentations: phones, PSC and syllables. The first two segmentations are deterministic in that only the syllabification needs to be known to derive lower-level segmentations. The chosen unit for pronunciation modeling has traditionally been the phone, where the prediction of a confusion is conditioned on the canonical phone and its surrounding context. Following our theory in Sec. 2, however, the motivation of simulating a speaker's L1.5 unit inventory is to allow for multiple segmentations, thereby encoding varying amounts of context. In addition, we want to model augmentative linguistic subword features of a word for a richer decision context. We therefore propose to exploit a hierarchy of linguistically motivated units and features in a so-called Subword Parse Table (SPT).
Our proposed framework is inspired by Seneff et al. [8] who arranged various subword units in a parse table for a number of tasks. Our setup differs in that we assign both a linguistic class and a phonetic realization to every cell of the table, and also in that we generally consider different segmentations. Fig. 2a shows a SPT for the English word *'communication'*. Each horizontal layer represents a segmentation of phone clusters, where the lower part of a cell is the phone realization and the upper part is its class label. Classes effectively encode linguistic features inherent to the subword segmentation, e.g., syllable structure or position within the syllable or word.

B. Statistical Model

1) Training: Given a word with its *canonical* unit sequence $\mathrm{U} = \mathrm{u}_1, \ldots, \mathrm{u}_N$, the task of a pronunciation model is to predict an alternative *confusion* unit sequence $\hat{\mathrm{U}} = \hat{\mathrm{u}}_1, \ldots, \hat{\mathrm{u}}_M$ based on some training data. For now let us assume that no insertions or deletions occurred, thus $N = M$. Due to sparseness of word-level exemplars the model approximates the word pronunciation likelihood with an independence assumption of subword confusions:

$$p(\hat{\mathrm{U}}|\mathrm{U}) \approx \prod_{i=1}^{N} p(\hat{\mathrm{u}}_i|\mathrm{u}_{i-1}, \mathrm{u}_i, \mathrm{u}_{i+1}), \quad (1)$$

where u_i and $\hat{\mathrm{u}}_i$ are the canonical and confused phones at i, and $\hat{\mathrm{u}}_i$ is predicted based on its canonical left and right context. For $\hat{\mathrm{u}}_0$ and $\hat{\mathrm{u}}_N$ the left and right symbols, respectively, are word boundary markers. Because the term $P(\hat{\mathrm{u}}_i|\mathrm{u}_{i-1}, \mathrm{u}_i, \mathrm{u}_{i+1})$ is word independent we can score pronunciations of unseen

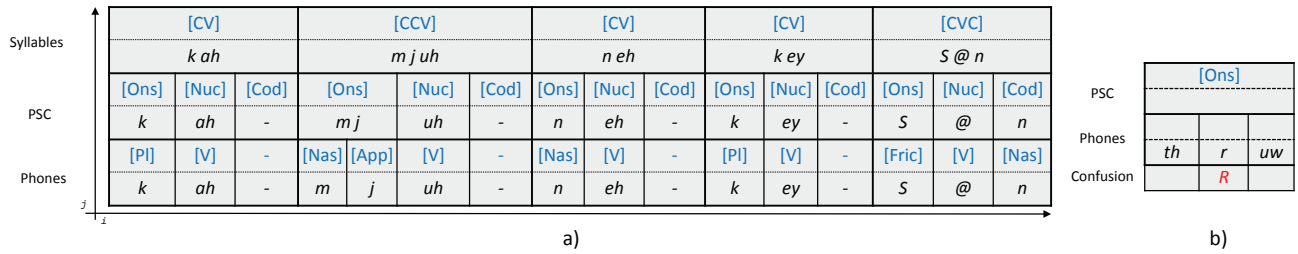

Fig. 2. a) Subword Parse Table for 'communication'. Layers reflect different segmentations. b) extracted c-gram for a cell in the SPT.

words. The unit u is traditionally the phone. By using hierarchical SPTs we can extend this approach by using larger-span subword units and by conditioning on richer context. We make two approximation assumptions: first, treat the different segmentation layers in a SPT as separate statistical models. This means that one model predicts phone confusions, one PSC confusions and one syllable confusions. Second, as we model confusion on a subword level, condition only on the local context of the cell. Each cell has a parent cell on the layer above (except those on the topmost layer), as well as a right and a left neighbor cell. This means that we can extract a context-gram (c-gram) C for each cell. The context can be extracted arbitrarily, but due to data sparseness we limit it to adjacent cells. In this contribution we define C to be a tuple $(\mathrm{u}_{i,j}, \mathrm{u}_{i-1,j}, \mathrm{u}_{i+1,j}, \mathrm{u}_{i,j+1}, \mathrm{class}(\mathrm{u}_{i,j+1}))$, where $\mathrm{class}(\mathrm{u}_{i,j})$ is the class of the cell corresponding to $\mathrm{u}_{i,j}$. Certain context conditionings can be dropped to allow for flexible configurations. An example of an extracted c-gram is illustrated in Fig 2b where an English /r/ is substituted by a German /R/ in a given c-gram context. Note that in this case some elements of the tuple are neglected. Eq. 1 implies independence between subword confusions. We follow a similar strategy in that we predict confusions on layer i for each cell independent of the confusions of adjacent cells:

$$p(\hat{\mathrm{U}}_j|\mathrm{U}_j) \approx \prod_{i=0}^{M-1} p(\hat{\mathrm{u}}_{i,j}|\mathrm{C}_{i,j}). \tag{2}$$

A model can then be estimated for each layer using manual phone-level transcripts of training data. First we align the confusion phone sequence with the cells of the layer in question (see Sec. 4) and then use these alignments to estimate

$$p(\hat{\mathrm{u}}_{i,j}|\mathrm{C}_{i,j}) = \frac{\mathrm{count}(\hat{\mathrm{u}}_{i,j}, \mathrm{C}_{i,j})}{\mathrm{count}(\mathrm{C}_{i,j})}. \tag{3}$$

Each layer encodes different linguistic features and varying context: on the phone layer, confusions are learned only for single phones and conditioned on short-span context, whereas the syllable layer models confusions based on syllable context.

2) Pronunciation Generation: For a given word and its SPT we want to generate a set of accented pronunciations, thresholded by δ. Remember that each layer is represented as a separate model for predicting confusions which means that we need to combine the scores obtained across all models for a given pronunciation. Consider the phone layer first. We start by generating all confusions for every cell, where each confusion is scored according to Eq. 3. The confusions and the canonical units of the cells can be represented as a lattice. The confusions of a cell are put on edges between two nodes, corresponding to binning in confusion networks. The posterior probability of a confusion corresponds then simply to the relevant edge score.

We generate lattices in the same way for the PSC and syllable layers. To interpolate the scores we need to align all three networks. This is trivial, since PSC and syllable edges can be broken into sequences of phone edges. Expanded phone edges retain the scores of their PSC or syllable edges they originated from, e.g., if $p(/\mathrm{s\ r\ uw}/|\mathrm{C}) = 0.8$ then $p(/\mathrm{s}/|\mathrm{C}) = 0.8$ etc.

From each of the three phone lattices, we can now calculate the posterior probability of a phone in a given bin, $p(\hat{\mathrm{p}}|\mathrm{bin}_j)$. For the phone lattice, this will be a given edge score, but for syllables or PSCs, multiple syllables and PSCs can have the the same phone at a bin position. We therefore sum over the scores of all edges that contain said phone. To interpolate the three scores for a confused phone $\hat{\mathrm{p}}$ we use

$$p(\hat{\mathrm{p}}|\mathrm{bin}_i) = \sum_{j=0}^{K-1} \alpha_j \cdot p(\hat{\mathrm{p}}|\mathrm{bin}_j), \tag{4}$$

where K is the number of lattices (or models), α_j is the interpolation weight for the j'th model and bin_i is the resulting composite bin. When merging the bins of the phone networks we assume that $\forall j < K : \hat{\mathrm{p}} \in \mathrm{bin}_j$. This, however, cannot be guaranteed since the phone layer encodes the least amount of constraint, and hence might generate phone confusions that were not observed on the PSC or syllable layers. In practice, we observed that, given our choice of interpolation weights, if the phone layer predicted a sequence that resulted in an unseen PSC or syllable, then resulting interpolated scores would be very small as PSC and syllable layers would just output a zero probability for that phone. We see this as future work, e.g., through a cascaded back-off structure.

3) Insertions & Deletions: Up to now we have assumed that the number of phones is the same for both the canonical and the predicted pronunciations. This is only true for substitutions, but not in case of insertions or deletions. A deletion can be treated straightforwardly as an empty substitution, i.e. $\hat{\mathrm{p}} = \epsilon$. As soon as units are inserted, however, the structure of the SPT and thus the number of bins in the network change. As we want to avoid reparsing the deterministic SPT, we simplify the problem by modeling insertions only on the phone layer. We then assume that an insertion is only dependent on the c-gram and the substitution of the phone cell to its left, and is independent of any higher-span layer. Eq. 4 is then extended to estimating the joint probability of a confusion along with a possible insertion $\mathrm{p}^\star$ to its right:

$$p(\hat{\mathrm{p}}, \mathrm{p}^\star|\mathrm{bin}_i) = p(\hat{\mathrm{p}}|\mathrm{bin}_i) \cdot p(\mathrm{p}^\star|\hat{\mathrm{p}}, \mathrm{C}_{i,0}) \tag{5}$$

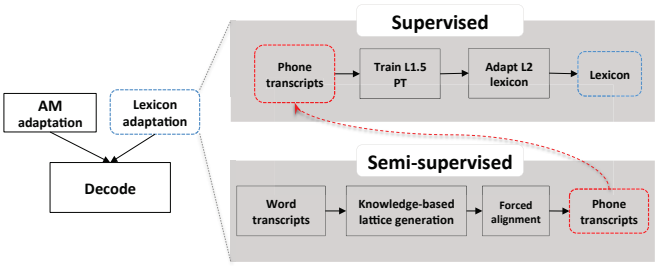

Fig. 3. Supervised and semi-supervised adaptation framework.

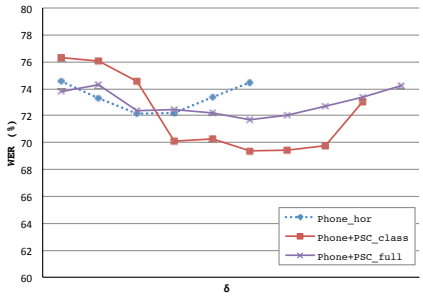

Fig. 4. Italian WER as a function of varying prediction threshold δ.

The idea is that we represent the edges of a bin as phone pairs, i.e. the confusion and the insertion, where the probability of the insertion is only dependent on the c-gram of the corresponding phone-level cell and the confusion phone itself. As insertions can be infrequent, we have to allow for ϵ insertions. To align a mispronunciation of w with the canonical phone sequence of the SPT's phone layer we use minimum edit-distance. Each cell at i of the phone layer is now associated with a phone tuple, where insertions are grouped to their previous phones (which can be canonical or a substitution). As a result we have the canonical pronunciation aligned with phone tuples, which can then be used to train the statistical model. For example, canonical /p aa r t/ and mispronounced /p aa r . d ax/ would align to /p→p,ϵ aa→aa,ϵ r→r,ϵ t→d,ax/. Consecutive insertions are merged into a single $\mathrm{p}^\star$.

4) Pronunciation Scoring: Confusions for higher-order layers are generated by taking the daughter cells of the cell in question and merging their confusion phones, e.g., the confusions of the Onset, Nucleus and Coda cells are merged to generate the confusion for a syllable cell. Note that canonical pronunciations are treated as confusions. By traversing a sequence of edges $\hat{\mathrm{E}}$, where each edge is a tuple with $\hat{\mathrm{e}} = (\hat{\mathrm{p}}, \mathrm{p}^\star, p(\hat{\mathrm{p}}, \mathrm{p}^\star|\mathrm{bin}_i))$ a set of pronunciations can be generated (by concatenating the phone pairs of all edges) and scored with a modified Eq. 2:

$$p(\hat{\mathrm{E}}|\mathrm{SPT}) \approx \prod_{i=0}^{N} p(\hat{\mathrm{e}}|\mathrm{bin}_i) \tag{6}$$

IV. Adaptation Framework

In this section we present the framework for accented pronunciation adaptation. As depicted in Fig. 3 a language-specific L1.5 accent model can be trained either supervised or semi-supervised, where the latter produces the phone-level training data for the first. Parallel to lexicon adaptation we use standard AM adaptation techniques for speaker adaptation. To generate parse tables we require syllabified L1 and L2 pronunciation lexicons.

A. Supervised Lexicon Adaptation

Training the statistical model proposed in Sec. 3 requires phone-level transcripts of the training data. In the supervised training scenario, transcripts on both word and phone levels are available. In addition, a lexicon containing the syllabification for each training word is needed to build the SPT. For each training word w we build a deterministic SPT representation from the word's syllabification, align the mispronunciation with the SPT and update $p(\hat{\mathrm{p}}, \mathrm{p}^\star|\mathrm{bin}_i)$. We repeat this for all training exemplars. The training data is in form of speech transcripts which means that higher frequency words have more impact on the resulting confusion distribution. Given the trained L1.5 model we expand the ASR lexicon as described in Sec. 3, while we only include pronunciations with a sequence score (see Eq. 6) higher than threshold δ. We do not weight the variants in the lexicon with posterior probabilities.

B. Semi-supervised Lexicon Adaptation

One way of obtaining phone-level transcripts is through expensive expert annotation. Given some accented speech and a word-level transcript it is therefore desirable to generate the training data for supervised adaptation automatically. Most pronunciation modeling approaches use a form of open phone decoding [9] to generate an expanded confusion search space for a word. This, however, usually results in extremely high error rates for accented speech. Motivated by the accent model of Sec. 2 we follow a more knowledge-driven approach: given the SPT's phone segmentation we expand each phone with a set of *confusable neighbors* which a listener could produce instead of the canonical phone. The IPA table organizes phones across languages in a way that reflects proximity between articulators on the horizontal axis. We translate perceptually confusing a phone in L2 with a L1 phone into a movement between the cells of the IPA table, that is, a phone is perceived as a phone of an adjacent cell. As we do not have any prior knowledge of what confusions can actually occur between L1 and L2 we allow for moving to an adjacent cell in either direction (using both voiced and unvoiced phones). Although moving vertically is not motivated by closeness of articulation, we included that constraint for flexibility. For example, canonical phone /th/ would be expanded to /th, dh, s, z, t , d, S, dz, f, v, r/.

Besides IPA confusions we also allow for insertions and deletions. Again, deletions are straightforwardly modeled with ϵ-confusions. Insertions are more complicated: allowing for an insertion after every cell would result in too big decoding networks (without insertions, allowing for 5 expansions plus canonical for each cell, would result in 6^N possible pronunciations. With a potential insertion after every cell, we would end up with 12^N paths using only one phone type as insertion). We thus estimate the location of an insertion by considering the phonotactics learned from the speaker's L1. If a L2 PSC is not in L1, we observed from data that the speaker often breaks the invalid cluster by inserting a vowel, thus forcing the pronunciation to agree with his internal phonotactic model. For example Spanish tends to not have plosives syllable final. The uni-syllabic word /p aa r t/ ('part') is likely to be split into /p

TABLE II
RECOGNITION ACCURACIES (%). 'MODELS' DESCRIBES WHICH LAYERS OF THE SPT WERE USED AND 'C-GRAM' DENOTES THE CONDITIONING CONFIGURATION. δ DENOTES THE SCORING THRESHOLD.

Models	Lang	c-gram	δ	Corr	Sub	Del	Ins	WER
base	Ital	-	-	43.2	52.7	4.1	19.6	76.4
base_2pron	Ital	-	-	43.0	53.3	3.7	21.1	78.0
Phone	Ital	left/right	0.2	46.9	49.4	3.8	19.0	72.2
Phone+PSC	Ital	full	0.03	47.3	49.0	3.8	19.0	71.7
Phone+PSC_noIns	Ital	class	0.03	47.1	48.2	4.6	17.8	71.9
Phone+PSC	Ital	class	0.03	47.4	48.0	4.6	16.8	69.4
base	Ger	-	-	53.3	40.1	6.6	7.8	54.5
base_2pron	Ger	-	-	52.5	41.8	5.7	9.1	56.6
Phone	Ger	left/right	0.3	55.6	39.3	5.1	8.9	53.3
Phone+PSC	Ger	full	0.1	55.6	39.4	5.0	9.1	53.6
Phone+PSC_noIns	Ger	class	0.06	55.3	39.1	5.7	8.5	53.2
Phone+PSC	Ger	class	0.06	55.3	39.0	5.7	8.4	53.0

aa r . t ax/ such that the plosive /t/ becomes the Onset of a new syllable. As /r t/ in Coda position is an OOU in the speaker's L1 unit inventory but /r/ is not we assign an insertion phone to /t/'s phone cell. The idea is that we break-off the phone that causes a PSC to be OOU by grouping the phone with a vowel. We restricted the insertion phone to schwas (/ax/) only.

In the same way as we generate a lattice from a SPT we assign a set of IPA confusions and possibly an insertion for each phone cell of the SPT and produce a phone-level network for the given word. Since the word sequence of an utterance is known we combine word-level networks into a decoding network for the whole utterance for forced-alignment. The resulting 1-best phone hypotheses are then segmented into word pronunciations and used as training data for supervised training.

V. EXPERIMENTAL SETUP

We train our SPT models and evaluate the adaptation framework on the ISLE database [10]. The corpus consists of 18h of Italian and German accented English speech read by 23 speakers of each language, where half of the data for each language is annotated on a phone-level. We use the phone-annotated data as training set and the rest as the evaluation set which corresponds to the originally defined split. The phone-level transcript of the training data is used to train pronunciation models for German and Italian. We generate English SPTs for the ISLE training words using a large syllabified lexicon based on CELEX [11]. For LVCSR decoding we use a 40k word recognition lexicon (there are 700 unique word types in the ISLE corpus, thus no OOVs) with one or two pronunciations per word, `base` and `base_2pron` respectively. Two 72-mixture triphone AMs were used: both trained with HLDA and VTLN, but one using ML and the other MMI. Both used 2000h of Switchboard and Fisher data with a base 3-gram LM trained on a mixture of telephone conversations, broadcast news and lectures. Acoustic adaptation was done using a cascade of global MLLR adaptation followed by 256-class regression tree MLLR adaptation. The acoustic data for a specific speaker from the ISLE training set was used to generate an adapted speaker-dependent model for that speaker. We use Microsoft Research's LVCSR engine for word decoding and HVite for forced-alignment. The ML AMs were used for forced-alignment. Results are reported on Word Error Rate (WER).

TABLE III
WER (%) OF BASELINE AND ADAPTED LEXICONS USING VARIOUS AMS.

Dict	AM adapt	Lex adapt	Italian	German
base	ML	-	76.4	54.5
Phone+PSC_class	ML	sup	69.4	53.0
Phone+PSC_class_noIns	ML	semi-sup	70.8	52.8
Phone+PSC_class	ML	semi-sup	69.4	52.7
base	DT+VTLN	-	73.7	51.0
Phone+PSC_class	DT+VTLN	sup	65.8	47.7
base	DT+VTLN+MLLR	-	67.8	42.1
Phone+PSC_class	DT+VTLN+MLLR	sup	61.3	39.7
Phone+PSC_class_noIns	DT+VTLN+MLLR	semi-sup	64.0	40.2
Phone+PSC_class	DT+VTLN+MLLR	semi-sup	62.6	40.1

VI. RESULTS

A. Supervised Adaptation

Table 2 reports results for supervised lexicon adaptation with baseline AMs on the test set.

Baseline: Italian accented English is a considerably harder ASR task compared to German accented English. Across both languages using pronunciation variants which tend to be helpful for native speakers result in increased WER. The insertion rate for Italian is higher because Italians insert vowels to avoid closed syllables, which in turn leads to an increase in hypothesized words in the search space.

Best systems: We adapt `base` with different configurations of the framework: we first vary the number of subword models in Eq. 4, where `Phone` denotes only using the phone layer and `Phone+PSC` denotes interpolating the both phone and PSC layers. Second, we use various amounts of conditioning context included in the c-gram, namely full c-gram context `full`, only the horizontal elements of the tuple `left/right` and only the class of the above cell `class` without left/right context.

The setup using `Phone` with `left/right` is the same as the traditional phone re-writing approach [2], only augmented with an insertion model, and yields 4.3% and 1.2% abs. improvement for Italian and German respectively. Using `full` c-grams (i.e. horizontal as well as class conditioning) on the phone layer did not give any improvements (and thus no numbers are reported). This means that the PSC class does not give any additional constraint over the canonical left/right phone information. Using `Phone+PSC` while conditioning on the full context, no significant improvement can be achieved over the `Phone` model either. However, restricting the c-gram to `class` (i.e. class conditioning only) gives overall gains of 7.0% and 1.5% abs. over the baselines on Italian and German, respectively. The additional knowledge appears to improve the robustness of the overall scores, yet when adding more horizontal context on the PSC layer the estimates are likely to be too sparse which leads to poorer scoring.

We do not report results for using the syllable layer since a missing back-off strategy prevents us from using the sparse evidence from syllable confusions. Fig. 4 shows the effect of varying threshold δ on Italian WER. For all three dictionaries WER drops initially after adding the most likely mispronunciations. Too many additional variants increase confusability with other irrelevant pronunciations, resulting in increased WER.

Insertions: For Italian the insertion rates decrease with improved scoring. When not modeling insertions the best Italian configuration's WER increases by 2.5% abs., while on German a slight decrease of 0.2% abs. is observed. A reason for this

could be that German's syllable structure is closer to English which makes the insertion model less important.

Speaker dependency: As the same speakers are both in the test and training parts of the data, we were concerned that our pronunciation models were overtrained on idiosyncrasies. Therefore we removed a speaker's speech from the training set, trained a model and ran an evaluation. Averaged across all speakers, the WER increased only slightly by 0.2% on Italian, meaning that the models are able to generalize.

AM adaptation: Results obtained with different AMs are reported in Table 3. Using unadapted AMs of improved quality with the baseline lexicon decreases WER by 2.7% and 3.5% abs. for Italian and German, respectively. Using `Phone+PSC_class` gives more improvement than using the base AMs, namely 7.9% and 3.3% abs.. Especially for German this could mean that improved AMs are able to better discriminate between confusable pronunciations added during lexicon adaptation. Using speaker adapted AMs yields 5.9% and 8.9% abs. improvements with the baseline lexicon, and overall 12.4% and 11.3% abs. with adapted lexicons. Thus, acoustic and lexical adaptation model complementary sources of variation.

Overall, the proposed lexicon adaptation is able to learn pronunciation patterns from training data and generalize to unseen words. Gains are consistent across both languages and the best configuration proved to be using both phone and PSC layers with no horizontal context in combination with adapted AMs.

B. Semi-supervised Adaptation

Generating the training data through knowledge-based forced-alignment and training the pronunciation model on this data produces similar or even better error rates compared to manually transcribed training data, seen in Table 3. Predicting insertions based on the L1 phonotactics appears to result in similar gains as learning insertions from supervised transcripts. The improvements in combination with AM adaptation are smaller compared to using baseline AMs on Italian (5.2% vs. 7.0% abs.) but similar on German.

AM confusion: When inspecting the pronunciations produced by forced-alignment some phone confusions (restricted by IPA movements) were not due to mispronunciations, but rather due to AM confusability, especially for dental fricatives /th/ and /dh/ which were chosen over /t/ or /d/ (the other way around would have been more sensible). This means that some of the gains were due to explicit AM confusion modeling in the lexicon. We compared the best supervised and semi-supervised dictionaries on German by creating sub-dictionaries containing 1) variants only in supervised 2) variants only in semi-supervised and 3) the union of both dictionaries. Supervised- and semi-supervised-only yield 54.2% and 53.5% WER. The union results in 52.4% which means that both approaches model different pronunciations that decrease WER. Another reason for performance difference is inconsistently annotated supervised data: the English and German /r/s are inherently different, yet are labeled with the same symbol in ISLE.

To further validate that semi-supervised adaptation is not only reducing AM confusion we trained SPTs based on a native English version of the ISLE prompts (6 speakers, 2h of speech). Using the baseline lexicon results in a WER of 24.0%, which is substantially better compared to recognizing non-native speech. Using two native pronunciation variants per word (`base_2pron`) decreases WER by 1.5% abs. (22.5%) which, compared to the non-native case, is in line with prior findings. We achieved similar gains with semi-supervised modeling (22.4%), meaning that a native pronunciation lexicon can be optimized given some native training speech.

VII. Conclusion

We proposed to use multiple subword segmentations to learn mispronunciations in both supervised and semi-supervised approaches. To improve robustness against non-native accent, we presented and evaluated an LVCSR framework which combines lexicon and AM adaptation. Motivated by a model of human speech processing, we found that combining constraints from phone and PSC layers gave best results. Only adapting the lexicon resulted in up to 7.9% and 3.3% absolute improvements for Italian and German, respectively. Combined with AM adaptation, the error rate could be reduced by absolute 12.4% on Italian and 11.3% on German. In comparison, [9] reports gains of 4.6% abs. on WER on the Italian part of ISLE when adapting a lexicon with only the 700 in-corpus words. Using semi-supervised lexicon adaptation, we achieve 7.0% abs. with a similar configuration adapting a 40k lexicon. Although these results are promising the reasons for the performance gap between native and non-native ASR remain not fully understood.

The appealing aspect of the framework is that additional linguistic subword features, such as morphology, graphemic knowledge or stress, can easily be integrated in a parse table. Using L1 and L2 unit differences could also be exploited to isolate potential error-prone areas in automated pronunciation assessment. Due to sparseness of certain features, however, a back-off mechanism is needed to improve score combination. When generating the training data in semi-supervised adaptation, more L1 knowledge could be incorporated, in form of L1 AMs and L1 LTS constraints. Future work will also focus on iterative lexicon and AM adaptation.

References

[1] C. Teixeira, I. Trancoso, and A. Serralheiro, "Recognition of non-native accents," in *Proc. Eurospeech*, 1997, pp. 2375–2378.

[2] H. Strik and C. Cucchiarini, "Modeling pronunciation variation for asr: A survey of the literature," *Speech Communication*, vol. 29, no. 2-4, pp. 225–246, 1999.

[3] H. Kim, M. Kim, and Y. Oh, "Non-native pronunciation variation modeling for automatic speech recognition," *Advances in Speech Recognition, InTech*, 2010.

[4] S. Greenberg, H. Carvey, L. Hitchcock, and S. Chang, "Beyond the phoneme: A juncture-accent model of spoken language," in *Proc. HLT*, 2002, pp. 36–43.

[5] J. Flege, "Second-language speech learning: Theory, findings, and problems," in *Speech Perception and Linguistic Experience: Issues in Cross-language research.*, W. Strange, Ed. Timonium, MD: York Press, 1995, pp. 229–273.

[6] K. Livescu and J. Glass, "Feature-based pronunciation modeling for speech recognition," in *Proc. HLT*, 2004, pp. 81–84.

[7] T. Mertens, D. Schneider, and J. Köhler, "Merging Search Spaces for Subword Spoken Term Detection," in *Proc. Interspeech*, 2009, pp. 2127 – 2130.

[8] S. Seneff, "The use of subword linguistic modeling for multiple tasks in speech recognition," *Speech communication*, vol. 42, no. 3-4, pp. 373–390, 2004.

[9] T. Tan and L. Besacier, "Improving pronunciation modeling for non-native speech recognition," in *Proc. Interspeech*, 2008, pp. 1801–1804.

[10] W. Menzel, E. Atwell, P. Bonaventura, D. Herron, P. Howarth, R. Morton, and C. Souter, "The isle corpus of non-native spoken english," in *Proc. LREC*, 2000, pp. 957–963.

[11] R. H. Baayen, R. Piepenbrock, and v. Rijn, "The CELEX lexical data base on CD-ROM." Philadelphia, PA: Linguistic Data Consortium, 1993.