Using Acoustics to Improve Pronunciation for Synthesis of Low Resource Languages

Sunayana Sitaram, Serena Jeblee and Alan W Black

Carnegie Mellon University, USA

ssitaram@cs.cmu.edu, sjebee@amazon.com, awb@cs.cmu.edu

Abstract

Some languages have very consistent mappings between graphemes and phonemes, while in other languages, this mapping is more ambiguous. Consonantal writing systems prove to be a challenge for Text to Speech Systems (TTS) because they do not indicate short vowels, which creates an ambiguity in pronunciation. Special letter-to-sound rules may be needed for some cases in languages that otherwise have a good correspondence between graphemes and phonemes. In the low-resource scenario, we may not have linguistic resources such as diacritizers or hand-written rules for the language. We propose a technique to automatically learn pronunciations iteratively from acoustics during TTS training and predict pronunciations from text during synthesis time. We conduct experiments on dialects of Arabic for disambiguating homographs and Hindi for discovering the schwa-deletion rules. We evaluate our systems using objective and subjective metrics of TTS and show significant improvements for dialects of Arabic. Our methods can be generalized to other languages that exhibit similar phenomena.

Index Terms: speech synthesis, low resource languages, lexicons, homographs, pronunciation

1. Introduction

Text to Speech (TTS) systems either make use of a lexicon to look up the pronunciation of words or use letter-to-sound rules trained on a lexicon. In many languages of the world the correspondence between the written form and the pronunciation is fairly straightforward. In some others, this relationship can be more complex, and sometimes ambiguous.

In consonantal writing systems like Arabic and Hebrew, the diacritics that indicate short vowels are usually omitted. This creates ambiguity when it comes to pronunciation, which native speakers deal with by looking at the context of the word. Without knowing the diacritics, it is not possible to disambiguate the pronunciation of this word in isolation. This can be a challenge for TTS systems particularly in the case of low resource languages in which tools like POS taggers may not be available or be very accurate. Also, some of these variants may not be predictable by context or POS but may need a deeper understanding of the semantics of the utterance.

In Indo-Aryan languages such as Hindi, the letter-to-sound mapping is quite straightforward, but there are special cases of deletion of schwas at the middle and end of words. These rules need to be incorporated into the front end of a TTS system to sound natural, like a native speaker of Hindi would sound. Typically, such rules are written by hand or trained from labeled examples.

In this paper, we propose a technique to use acoustics to disambiguate pronunciations in Arabic using acoustics. We build Text to Speech systems using the pronunciations that the acoustic model selects iteratively until an objective measure of TTS system quality converges. At synthesis time, we predict the pronunciation of the word by looking at linguistic features and context.

We use a similar technique to automatically come up with schwa deletion rules for Hindi, using only acoustic features from recordings of TTS databases. Although many attempts have been made at coming up with schwa deletion rules for Hindi, this phenomenon exists in various other Indian languages, where such attempts have not been made, and for low resource languages, such automatic grapheme-to-phoneme rule induction techniques may be useful. In both the above problems, we deal with languages that have well defined writing systems. Our techniques can be applied to languages that do not have their own standardized writing system [1].

The rest of the paper is organized as follows. Section 2 relates this paper to prior work. Section 3 describes the data and resources we used for our experiments. Sections 4 describes the pronunciation choice experiments for dialects of Arabic. Section 5 describes the experiments for learning letter to sound rules in Hindi. Section 6 concludes.

2. Relation to prior work

The problem of pronunciation choice for homographs noted above that is particular about certain languages also appears to some degree in all languages. In earlier work we have considered how speaker (and style) lexical choices affect synthesis using both acoustic models to label them [2] [3] and statistical models at synthesis time to choose the right variant. That work was targeted at very localized choices, such as vowel reduction but we felt that technique could scale up to this problem.

Also this work has some similarity to the early work of Yarowsky [4] on homograph disambiguation. However here we do not require any human labeling of initial examples, but rely on the acoustic models to find these variants in the data. But then, like Yarowsky, we predict which distinct homographic instance to use as both training time (to improve our models) and at test time when doing novel synthesis.

Data driven techniques have been used to model pronunciation variation for ASR by making use of acoustically derived subword units [5] by using multiple speakers data pronuncing the same word. Knowledge based techniques make use of phonological rules to create more variants [6] to decrease the Word Error Rate of ASR.

SALAAM [7] is a technique in which an existing high quality ASR system is used to automatically generate pronunciations in a target language through cross-lingual phonetic decoding. These pronunciations are then used as the lexicon by the ASR system to decode speech in the target language. The main application of this method is to build an ASR system for the low resource target language, and the pronunciations are created to maximize ASR discrimination between them [8]. These pronunciations, however, may not be suitable for a TTS system to synthesize from. Also, the SALAAM method has been used for low vocabulary scenarios and needs multiple instances of training data for each word which are not necessarily available in a TTS database.

The schwa deletion phenomenon has been well studied in the context of Text to Speech systems. There are well defined linguistic rules to predict when a schwa is deleted and when it is not deleted. However, there are exceptions to these rules that are reported as being around 11% of the vocabularly [9]. With the addition of new words and foreign words, one would expect this number to be quite high. Also, Hindi and other Indian languages being low resource, there are no lexicons available that can be used to automatically train these rules from.

Previous work on schwa deletion includes approaches that take into account morphology [9] to preserve schwas that may otherwise be deleted. Other approaches have used syllable structure and stress assignment to assign schwa deletion rules [10]. [11] uses ease of articulation, acoustic distinctiveness and ease of learning to build a constrained optimization framework for schwa deletion.

Our approaches to both these problems involve using the acoustics, or the way the voice talent pronounced words in order to learn more about the pronunciation of words, and try to generalize them into rules that can be used for new words.

3. Data and resources

First, we will describe the resources we used for our experiments on choosing pronunciation variants, followed by details about the resources used for the LTS rules experiments.

3.1. Choosing pronunciation variants

We applied our techniques to three databases in two languages two in Iraqi Arabic and one in Modern Standard Arabic (MSA).

3.1.1. TTS databases

We used data from BBN created for the DARPA BOLT (Broad Operational Language Translation) Program and Iraqi Arabic TTS data from the DARPA TRANSTAC program for the Iraqi Arabic TTS systems. The BBN data had 62 minutes of speech from a male speaker. The Transtac data had 74 minutes of speech from a male speaker. In both cases, the corresponding transcripts did not have any diacritics.

For Modern Standard Arabic (MSA), we used data from the SASSC [12] corpus. The SASSC database contains single male speaker data spoken with different styles such as normal, funny, sad, questions etc. We used 50 minutes of data from the normal speech part of the database as our TTS data. The corresponding transcript was fully diacritized. We ran a script to remove the diacritics from the transcript.

3.1.2. Acoustic Models

For building an Iraqi Arabic Acoustic Model, we used 2-way dialogues between native Iraqi Arabic speakers, interpreters and native English speakers from the Transtac project. We extracted 20 hours of Iraqi Arabic utterances spoken by native speakers from the dialogues using manually annotated timestamps and transcripts. The transcripts did not contain any diacritics.

For building the MSA Acoustic Model, we used the rest of the normal speech in the SASSC corpus, leaving out the utterances labeled traditional which were in Classical Arabic. This came to around 5 hours of speech data. We removed the diacritics from the transcripts used for training the Acoustic Model.

For both Acoustic Models, we used the default (first) pronunciation from the lexicon while building the models. We used the CMU Sphinx speech recognition toolkit [13] to train and run the GMM-based Acoustic Models.

3.1.3. Lexicons

We used an Iraqi Arabic lexicon from LDC [14] which contained words with and without diacritics in Iraqi script, Buckwalter transliteration [15] for each word, syllable boundaries and Part of Speech (POS tags). We created a phone set by mapping the Buckwalter characters to individual phonemes with the appropriate phonetic features. The LDC lexicon contains around 88k words, out of which 11k words have multiple pronunciations, with some words having as many as eight different pronunciation variants. We created a new lexicon using Iraqi Arabic surface forms without diacritics, phonemes with syllable boundaries and POS tags. We numbered the pronunciation variants in the lexicon as word, word(2), word(3) etc.

For MSA, we did not have a standard lexicon. However, we had the transcripts and labels for the acoustic data from the SASSC corpus. We aligned the phoneme labels with the words in the transcript and created a lexicon specific to our corpus. However, this alignment was not completely accurate. In Arabic, the determiner 'al' is often blended with the end of the previous word to create fluid speech. This makes determining the word boundaries difficult because the final vowel is combined with the initial 'a' of the 'al', and the determiner sounds like it is attached to the previous word instead of the correct word. This creates many words with extra phonemes at the end, and definite words that do not have the phonemes for the determiner. This results in lexicon entries that are not entirely correct.

In order to better process the MSA data, we removed all diacritics and normalized certain consonants that have many common variations (such as). To get part of speech tags for MSA, we used the Stanford Tagger[16] with their standard Arabic model. For Iraqi we used CALIMA [17], an morphological analysis tool for Arabic dialects developed at Columbia University.

3.2. Letter to Sound Rules

For discovering schwa deletion rules automatically from acoustics, we used Hindi as the low resource language.

3.2.1. TTS databases

Our Hindi data came from the 2014 Blizzard Challenge [18] and consisted of around an hour of speech from a professional male speaker.

3.2.2. Acoustic Models

Since we treated Hindi as the language with low resources, we built an acoustic model with data from other Indic languages. We created a corpus of an hour each of Bengali, Telugu, Tamil, Telugu and Rajasthani TTS data from the Blizzard Challenge and treated all these languages as higher resource languages. So, in order to label the data for building the acoustic models, we used the Festvox [19] Indic front-end grapheme-to-phoneme

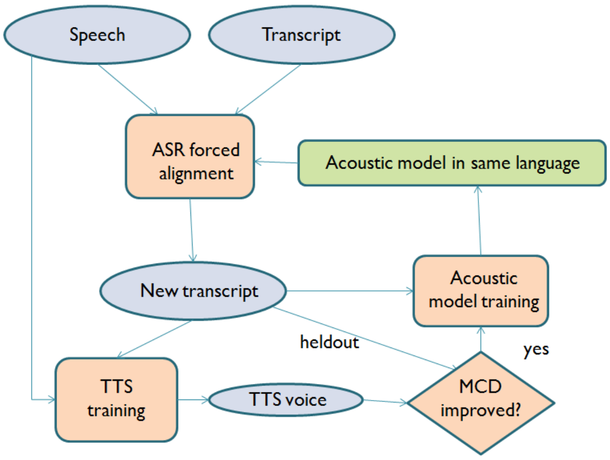

Figure 1: *Discovering pronunciations from acoustics.*

rules available with the standard distribution. As before, we used Sphinx for building and running the acoustic models.

3.2.3. Baseline and Knowledge-based LTS

For our baseline systems, we used UniTran [20] to get a mapping between the Unicode characters in Hindi and X-SAMPA phonemes. For our knowledge-based hand-written LTS rules, we used the Festvox Indic front end.

3.3. Voice building

We used the Festival speech synthesis engine [21] and the Festvox voice building tools [19] to build all our TTS systems for both sets of experiments. We built CLUSTERGEN [22] Statistical Parametric Synthesis voices during training to be able to obtain the Mel Cepstral Distortion (MCD) [23], an objective measure of TTS system quality.

4. Modeling pronunciation choice

4.1. Learning pronunciation choice from acoustics

Given a pronunciation dictionary with multiple pronunciations for ambiguous words, a well recorded TTS database and the corresponding transcript, our task is to choose the correct pronunciation from the lexicon for the words in the transcript. Figure 1 illustrates our basic approach to learning pronunciations from acoustics. First, we use an acoustic model from our target language to force align the original TTS transcript with the TTS speech data. During the process of forced alignment, the model chooses a pronunciation from the lexicon for a particular word based on the phonemes in the word and we get a transcript with word variants that are different from the original transcript.

Next, we use the new transcript and the TTS speech data to rebuild a small, targeted acoustic model. This targeted acoustic model is then used to force-align the transcript used for training the data. During forced alignment, the model chooses the most appropriate pronunciation from the lexicon that match the phoneme labels it creates during labeling. We repeat this process iteratively and at each stage build a CLUSTERGEN voice. We use the same held out set of sentences for all the iterations to test the TTS system and measure the MCD. We stop the iterations when the MCD no longer improves. At each stage, an alternative to building new targeted acoustic models is to do some kind of model adaptation, but we did not do so.

In previous work, we have seen that repeating this process iteratively while building cross lingual phonetic TTS systems for languages without a written form has given us gains, typically in the third or fourth iterations [1]. In these experiments, we observed a sharp decrease in MCD in Iteration 1 for all three TTS databases. In case of the Iraqi BBN database, there was a slight improvement in the MCD in Iteration 2.

Table 1: *Baseline and best iteration MCD scores*

Database	Baseline	Best Iteration
Iraqi BBN	4.67	4.21
Iraqi Transtac	4.88	4.35
MSA	6.64	6.34

Table 1 shows the MCD of the baseline system and the best iteration. In all three cases, we get a significant improvement in MCD compared to the baseline. An improvement of 0.08 in MCD is considered to be perceptually significant and an improvement of 0.12 is equivalent to doubling the training data [24].

4.2. Predicting pronunciations from text

At synthesis time, we need to be able to predict pronunciations from text. The Iraqi Arabic LDC lexicon contained Part of Speech tags for all the words, which we used to build Classification and Regression Trees (CART) to predict the pronunciation of a particular word, given its POS and the POS of the previous two and next two words. We used the Edinburgh Speech Tools [25] CART tree building program to build and test our trees. We built individual CART trees for each word and tested our trees on held-out data.

A survey of the incorrectly predicted words showed that most of the failed disambiguations were due to homographs with different pronunciations where contextual information is needed in order to choose the correct one, and only using POS was not enough. No reliable dependency parsers exist for Iraqi Arabic, so we used lexical features from the surrounding words to help with disambiguation. We also used induced POS [26] for MSA, but since we had a small amount of training data, we did not get reliable tags. We use the Iraqi version of CALIMA for morphological analysis. The morphological analysis includes a stem for each word, and we use this to extract prefixes, a stem, and suffixes for each word. The lexical feature vector consists of the stem and affixes for the target word as well as the next and previous word.

We found that the accuracy of the CART trees by performing 10-fold cross validation for Iraqi Arabic was very high at 93%, while for MSA it was much lower at 76%, which can be explained by the problems with the accuracy of the lexicon mentioned earlier and less training data for the trees.

4.3. Subjective evaluation

Previously, we saw that our iterative method resulted in better labeling for the three databases and hence better MCD, which resulted in much better quality that was perceptually significant. However, we wanted to test how good our predictions were and whether subtle variations in pronunciation could be perceived by native listeners.

We used the Testvox [27] tool for creating AB preference tests with a 'no difference' option for all our subjective tests.

In our first set of listening tests, we conducted preference tests with four native Arabic speakers outside our research group for Iraqi Arabic. Subjects were asked to listen to two synthesized sentences from the test set that had one word that our model predicted a different pronunciation for than the default pronunciation in the lexicon. We found that there was a slight preference for the utterances with our predictions compared to the baseline. In many cases, the difference in pronunciation was subtle and was not perceived in the listening test.

We thought it would be useful to have the subjects explicitly focus on the word that was different in the two utterances so that we could judge whether the pronunciations we predicted were different or not. We conducted subjective tests for MSA, and synthesized sentences similarly as we did for Iraqi. In the listening tests, we showed subjects the transcript with the ambiguous word highlighted and asked them to pick the synthesized utterance in which the word sounded correct, or choose a third option if they couldn't tell a difference. Table 2 shows the results of the listening test with nine native Arabic subjects, with each subject listening to ten pairs of sentences.

Table 2: *Subjective evaluation of pronunciation choice prediction for MSA*

Prefer Baseline	Prefer Predicted	No difference
4.44%	64.44%	31.11%

Our results for MSA show a significant preference for the predicted pronunciations compared to the baseline. The important thing to note here is that the rest of the sentence except the highlighted word was identical, and all sentences were synthesized with the same system using our best training labels. This result is very encouraging as it shows that the difference in pronunciation can be perceived and that we are making the right predictions.

5. Learning Letter-to-Sound rules

First, we wanted to see the impact schwa deletion has on the overall quality of the database. Our baseline system for Hindi used the UniTran frontend, which automatically assigned schwas to all consonants that did not have a vowel following them. We also built a system using the Indic front end, which had hand-written rules for schwa deletion. We found that the difference between the MCD of the two voices was 0.05, which is not very significant. In informal listening tests, we found that the difference between these voices was easily perceivable by native speakers. This indicates that MCD averaged across the entire database may not be the most appropriate metric to capture this phenomenon, considering that it occurs in around 40% of the words in the Blizzard Hindi corpus.

We used Assamese data to build two classes for the CART tree: a positive class with the correct schwa labels from the knowledge based Indic front end, and a negative class with spurious schwas from the UniTran baseline. We used the score that Sphinx assigns for each phoneme during forced alignment, that we will refer to as the Acoustic Score and its right and left contexts, and the duration of the phoneme and context phonemes as features in our model. So the idea was to look at these features in the synthesized speech from the UniTran baseline for Hindi after force aligning it with the Indic acoustic model, and predict if a schwa belonged to a negative or positive class, that is, whether it should be deleted or not. It should be noted that Assamese only contains word-final schwa deletion, but we hoped to capture word-medial schwa deletion in Hindi as well with this model.

After running the model, we got predictions about whether schwas should be deleted or not, in the Hindi data. We manually labeled 400 words to calculate the precision of these predictions and found that the precision was slightly better than chance. However, we found that many words were labeled with the correct schwa deletion rules more often than they were labeled wrong. So, we took the most frequent label of a word and created a new lexicon for Hindi with it. We used this lexicon to build a voice for Hindi.

We synthesized 10 sentences for Hindi and asked 10 native speakers of Hindi to choose between the systems built with the UniTran baseline and our predicted schwa deletion lexicon. We asked them to pick the system they felt had better pronunciation, with the option to pick "no difference". Table 3 shows the results of the subjective evaluation.

Table 3: *Subjective evaluation of schwa deletion*

Prefer Baseline	Prefer Predicted	No difference
30%	59%	11%

We can see that there was a preference for the system with the predicted schwa deletion rules, when compared to the baseline. Here, we used the UniTran mappings as the baseline, which does not delete the schwa at all, but we could also have used a random baseline, which randomly deletes schwas.

In this experiment, we only used the Acoustic Model confidence and duration as acoustic features in our CART trees. We are currently investigating other features, such as articulatory features and Inferred Phonemes [28]. In addition, we are also investigating better objective and subjective metrics that capture subtle differences in pronunciation.

6. Conclusion

In this paper, we proposed techniques to model pronunciation using speech from TTS databases. The idea behind this approach was to use the way the speaker pronounced words to learn more about the letter-to-sound mapping for the language. We applied this to two problems: pronunciation choice in dialects of Arabic (MSA and Iraqi) and schwa deletion in Hindi.

Our techniques showed a significant improvement in objective measures of TTS quality for all three databases for dialects of Arabic. We built a model to predict pronunciations at synthesis time from text using POS, lexical and context features, and found significant preference for pronunciations selected by the model in subjective evaluations for MSA. The problem of missing diacritics in languages written in the Arabic script extends to languages like Urdu, Farsi, Kurdish etc. One can also imagine applying the same techniques to European languages with missing accents in the transcript.

We also used a similar technique for Hindi schwa deletion, by building a model of schwa deletion in Assamese and using Indic acoustic models. Our methods can be generalized to learning letter-to-sound rules for other languages. Future work in this direction includes using better acoustic features, carrying out more rigorous subjective evaluation and finding a better objective metric that captures subtle pronunciation differences in TTS systems.

7. References

[1] S. Sitaram, S. Palkar, Y.-N. Chen, A. Parlikar, and A. W. Black, "Bootstrapping text-to-speech for speech processing in languages without an orthography," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7992–7996.

[2] C. L. Bennett and A. W. Black, "Using acoustic models to choose pronunciation variations for synthetic voices." in *INTERSPEECH*, 2003.

[3] C. Bennett and A. W. Black, "Prediction of pronunciation variations for speech synthesis: A data-driven approach," *ICASSP*, 2005.

[4] D. Yarowsky, "Homograph disambiguation in text-to-speech synthesis," in *Progress in speech synthesis*. Springer, 1997, pp. 157–172.

[5] T. Spiess, B. Wrede, G. A. Fink, and F. Kummert, "Data-driven pronunciation modeling for asr using acoustic subword units." in *INTERSPEECH*, 2003.

[6] M. Wester, "Pronunciation modeling for asr–knowledge-based and data-derived methods," *Computer Speech & Language*, vol. 17, no. 1, pp. 69–85, 2003.

[7] F. Qiao, J. Sherwani, and R. Rosenfeld, "Small-vocabulary speech recognition for resource-scarce languages," in *Proceedings of the First ACM Symposium on Computing for Development*. ACM, 2010, p. 3.

[8] H. Y. Chan and R. Rosenfeld, "Discriminative pronunciation learning for speech recognition for resource scarce languages," in *Proceedings of the 2nd ACM Symposium on Computing for Development*. ACM, 2012, p. 12.

[9] B. Narasimhan, R. Sproat, and G. Kiraz, "Schwa-deletion in hindi text-to-speech synthesis," *International Journal of Speech Technology*, vol. 7, no. 4, pp. 319–333, 2004.

[10] T. Naim R and I. Nagar, "Prosodic rules for schwa-deletion in hindi text-to-speech synthesis," *International Journal of Speech Technology*, vol. 12, no. 1, pp. 15–25, 2009.

[11] M. Choudhury, A. Basu, and S. Sarkar, "A diachronic approach for schwa deletion in indo aryan languages," in *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, ser. SIGMorPhon '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 20–26.

[12] I. Almosallam, A. AlKhalifa, M. Alghamdi, M. Alkanhal, and A. Alkhairy, "Sassc: A standard arabic single speaker corpus," in *proceedings of 8th ISCA Speech Synthesis Workshop*, 2013.

[13] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler *et al.*, "The 1996 hub-4 sphinx-3 system," in *Proc. DARPA Speech recognition workshop*, 1997, pp. 85–89.

[14] D. Graff, T. Buckwalter, H. Jin, and M. Maamouri, "Lexicon development for varieties of spoken colloquial arabic," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, 2006, pp. 999–1004.

[15] T. Buckwalter, "Buckwalter arabic morphological analyzer version 1.0," 2002.

[16] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 173–180.

[17] N. Habash, R. Eskander, and A. Hawwari, "A morphological analyzer for egyptian arabic," in *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*. Association for Computational Linguistics, 2012, pp. 1–9.

[18] K. Prahallad, A. Vadapalli, S. Kesiraju, H. A. Murthy, S. Lata, T. Nagarajan, M. Prasanna, H. Patil, A. K. Sao, S. King, A. W. Black, and K. Tokuda, "The blizzard challenge 2014," 2014.

[19] A. W. Black and K. A. Lenzo, "Building synthetic voices," *Language Technologies Institute, Carnegie Mellon University and Cepstral LLC*, 2003.

[20] T. Qian, K. Hollingshead, S.-y. Yoon, K.-y. Kim, R. Sproat, and M. LREC, "A python toolkit for universal transliteration." in *LREC*, 2010.

[21] P. Taylor, A. W. Black, and R. Caley, "The architecture of the festival speech synthesis system," 1998.

[22] A. W. Black, "Clustergen: a statistical parametric synthesizer using trajectory modeling." in *INTERSPEECH*, 2006.

[23] M. Mashimo, T. Toda, K. Shikano, and N. Campbell, "Evaluation of cross-language voice conversion based on gmm and straight," 2001.

[24] J. Kominek, T. Schultz, and A. W. Black, "Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion." in *SLTU*, 2008, pp. 63–68.

[25] P. Taylor, R. Caley, A. W. Black, and S. King, "Edinburgh speech tools library," *System Documentation Edition*, vol. 1, 1999.

[26] A. Clark, "Combining distributional and morphological information for part of speech induction," in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 59–66.

[27] A. Parlikar, "Testvox: Web-based framework for subjective evaluation of speech synthesis," *Opensource Software*, 2012.

[28] P. K. Muthukumar and A. W. Black, "Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2594–2598.