EXTRACTION OF LINGUISTIC INFORMATION WITH THE AID OF ACOUSTIC DATA TO BUILD SPEECH SYSTEMS

Sebsibe Hailemariam †, Kishore Prahallad ‡†

International Institute of Information Technology (IIIT), Hyderabad (India) †
Language Technology Institute, Carnegie Mellon University (USA) ‡

ABSTRACT

Linguistic knowledge plays a major role in building high quality speech systems such as speech synthesizer and speech recognizer systems. Quality of such speech systems greatly depends on the availability of linguistic knowledge such as pronunciation dictionary, stress pattern, syllable structure and so on. However, it is very difficult to obtain these linguistic knowledge in the required format for most of the languages. In this paper an attempt has been made to design and develop tools for extraction of the core linguistic information (letter to sound rule) automatically with the aid of acoustic data to build improved quality speech systems. Acoustic evidences will be exploited with minimal language knowledge to deploy the vital linguistic resources for such systems. The resulting Letter To Sound (LTS) rule is tested on different test data on Amharic, Hindi and Tamil. The performance of the letter-to-sound rule is reported in this paper.

***Index Terms*—** Transcription correction, Grapheme-to-phoneme conversion, speech segmentation, and linguistic information for speech systems

1. INTRODUCTION

Linguistic information is vital resource for the success of speech synthesis and recognition. This information includes pronunciation dictionary, stress pattern, pause prediction, language model, intonation, and others. For languages which don't have such linguistic information, one way to obtain this resource is using the language expert(s) and generate the resource manually. Compilation of such resource in the required format and size takes time as well as requires large capital investment [1]. In some situations, it is even difficult to find an expert in the language area to drive manually the required information. Pronunciation dictionary is the minimal and core linguistic knowledge for current speech recognition and synthesis systems [2], [1], [3].

Pronunciation dictionary provides a means to map a word into its elementary phonetic components which is a key for modeling both speech recognition and synthesis systems. Naturalness of speech synthesis system highly depends on the intonation, pause and stress prediction. Similarly, research shows proper modeling of stress pattern and language modeling improves performance of speech recognition systems [4]. Most or all of such important linguistic information are not yet readily available to most of the languages of the world. Such vital resources are so meager and scanty when it comes to minority languages. The term minority language in the context of this research refers to languages which are not well researched and do not have sufficient linguistic resources to build speech systems [5]. This term is adopted from articles written by [5] [1].

The scarcity of such vital resource in minority languages initiates this research to find techniques and strategies on the possibilities to build and improve speech synthesis and recognition systems. In this regard, this research tries to address techniques and strategies where by speech synthesis and recognition system can be built and its quality would be improved for minority languages. Grapheme based speech synthesis and recognition system can be built using grapheme as a basic unit. However, lack of proper mapping from grapheme to the pronunciation of a word in grapheme based speech systems limits performance of the speech systems. Basic linguistic information extracted with the aid of acoustic data and minimal language knowledge could be used to improve the quality of grapheme based speech system.

This paper is organized into five sections. Section 2 outlines grapheme based approaches for speech synthesis and recognition systems. Section 3 explain transcription correction algorithm and its appropriateness for improvement of speech systems. Section four provides experimental results and analysis of the result. The last section is conclustion and recommendation.

2. GRAPHEME BASED SPEECH RECOGNITION AND SYNTHESIS SYSTEMS

The current speech synthesis system lacks both intelligibility and/or naturalness and doesn't satisfy ever increasing human needs [6]. Current development of speech synthesis and recognition systems are limited to developed languages (which can provide the necessary linguistic information in the required format). The beneficiaries of such a system are also only those who are able to understand the language in which the speech systems are developed. Considering the rest of the languages is believed to be helpful for speech technology [5].

Grapheme based approach for speech synthesis and recognition has been reported on [5], [1], [7], [8], [9], [2]. In grapheme based speech synthesis and recognition systems, grapheme is used as a basic unit which makes word to basic unit mapping a trivial task. Grapheme based speech systems gives quite better result to language like Spanish which has close grapheme-to-phoneme relationship [5]. Systematic context information exploitation has proved to be of great importance for both speech synthesis and recognition system [5], [1], [2].

We have made similar experiment (grapheme based speech synthesis) for Amharic language which is nearly phonetic. Perceptual test has been conducted on the grapheme based Amharic voice to evaluate the performance of the speech synthesizer which is developed using festival/festvox synthesis engine. This perceptual evaluation conducted on 5 test Amharic sentences selected randomly from the corpus using 6 native speaker of the language. The evaluator assign a number in the range from 0 to 5 to indicate the naturalness of the synthesized speech. 0 stands for very poor and 5 stands for best quality speech output. The average result of the experiment is $2.56(51.3\%)$.

Previous work in [10] has a performance of $2.91(58.2\%)$ from perceptual test results on the same corpus using the phonemes as a basic units and rule based manually derived grapheme-to-phoneme conversion.

Similarly, experiment on the performance of speech recognition on the same language shows 39% word error rate and 65% sentence error rate when grapheme is used as a basic unit. These error rates are decreased to 33% and 40.2% respectively when phoneme are used as the basic unit. This experiment is conducted using sphinx speech recognition system developed at Carnegie Mellon University.

Comparison of performance of grapheme and phoneme based speech recognition system as well as the perceptual evaluation result of the grapheme and phoneme based speech synthesis system discussed above shows phoneme based speech synthesis and recognition gives better quality over the grapheme based unit.

The main reason considered for the performance gap between phoneme based and grapheme based speech systems is that, the phoneme sequence is a better representation of sounds present in the speech signal than the grapheme sequence and thus provide better alignment. Improper speech segment alignment has a negative effect on the quality of speech synthesis and recognition system. Improper mapping is not only the problem of grapheme based speech systems but also the problem of phoneme based speech systems as speakers often don't speak a word in a sentence as intended by its lexical pronunciation. This is due to various reasons such as speaking rate, speaking style, dialect, conversational versus reading speech etc. These variation can not be captured by typical letter to sound rule generated either hand crafted or by machine learning models.

Given a transcription either in terms of grapheme sequence or phoneme sequence, this paper proposes an approach to capture the intended pronunciation of the speaker with the aid of the acoustic data and minimal language information. One could seek such minimal language information by asking a native speaker about the mapping process of grapheme to phoneme. Note that such information can be sought from none language expert native speaker of the language. For example, the mapping of the letter 'C' to /ch/ or /k/ can be sought by non native English speaker without much difficulty. This becomes much easier if the language is nearly phonetic.

In our approach such minimal language information is represented as alternate pronunciation of grapheme or phoneme in HMM frame work using skip arcs. Then the acoustic hints will be used as an evidence for selection of the best pronunciation unit at that acoustic segment. This approach is referred to as "transcription correction" and is discussed in detail in next section.

3. TRANSCRIPTION CORRECTION

3.1. *Overview of transcription correction*

Transcription correction is an algorithm used to map transcription of a given speech utterance into its phonetic sequence representation as realized in the acoustic waveform. The writing system of a nearly phonetic language shows limited conflict between the grapheme and the phonemic unit it represents. This feature of the language simplify the effort required for mapping transcription into phoneme sequence as per the acoustic information. Depending on the language, one could make various observations that gives a hint on the mapping process. The acoustic hint will be used as an evidence for selection of the best pronunciation unit at that acoustic segment.

Deletion, insertion and replacement are the three basic operations performed by the mapping function during transcription correction using the acoustic hint. Grapheme unit get deleted when the grapheme doesn't have acoustic representation in the speech. Insertion of grapheme is possible when the grapheme is represented in the acoustic data but not in the transcription. Replacement is a phenomenon where there is a mismatch between the acoustic representation in that segment and the corresponding grapheme at that point.

The architecture of the transcription correction algorithm doesn't require specific information about when or where a particular unit is replaced, deleted or inserted. In other words, context analysis is not as such important to determine if a particular sound that correspond to the grapheme is deleted or inserted or replaced. If the context information are obvious for the native speaker, then inclusion of that information will minimize the CPU time required for training. This strategy can be used even to non-phonetic languages with complex design issues which are purely engineering issue than linguistic.

The following are the general ideas we have used during transcription correction for the languages Hindi, Tamil and Amharic. The ideas are obtained from non-language expert native speaker. Hence the same can be done to all minority languages.

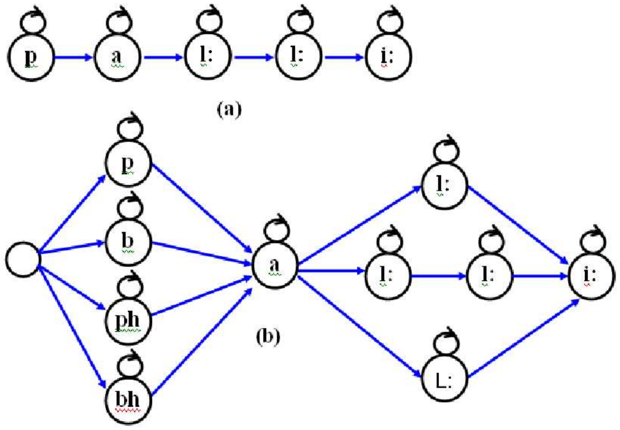
1. In all languages, prosodic pause or breath may be inserted at the end of each word.
2. In Hindi and Tamil two similar consonant can appear in sequence (gemination) but it may not be possible to have corresponding sound segment to each of them. Therefore we add a possibility of mapping the two consonant grapheme into one represented by either one of the two grapheme or a brand new sound unit representation.
3. In Hindi, the grapheme that represent schwa may or may not be deleted from the transcription.
4. In Tamil, some grapheme may have multiple pronunciation which can be modeled with multiple replacements.
5. In Amharic, consonant clusters may have epenthesis vowel inserted between them.
6. In Amharic some grapheme can be replaced with another grapheme.
7. In Amharic, some grapheme may not be pronounced.

The transcription correction algorithm takes such language rules as input to perform the mapping function.

3.2. Algorithm for transcription correction

This section describe the methods used to map acoustic feature vectors and grapheme sequence into acoustically motivated phoneme sequence. The mapping function uses Hidden Markov Model (HMM). A three state unit (grapheme and its variants) level HMM is defined to all units of the target languages. The units HMM will joined to each other according to their occurrence in the transcription to form word level HMM. The word level HMM get modified using the hints extracted from the language speaker given to the system as an input. This allows insertion of new units, deletion of existing units, and/or replacement of existing units with brand new units and so on. These word level HMMs further will get joined each other to form the sentence level HMM. Further modification of the HMM architecture will be made to reflect changes at word boundaries such as controlling of breath/pause, specific language rule to be made at the beginning of the next word or end of the current word.

Consider the Tamil word pal:l:i (school) with grapheme sequence [p], [a], [l:], [l:] and [i]. The language rule says, [p] can represent [p], [b], [ph], or [bh] sound. moreover, the clustered grapheme [l:] followed by [l:] can represent two [l:] sounds or single [l:] sound or stressed version of [l:] represented as [L:]. Figure 3.2 shows word level HMM description of the word pal:l:i. At the top (A), we showed HMM level description before applying modification rule and at the bottom, we showed HMM level description of the same word after applying modification rule.



Word level HMM description of the word pal:l:i

Transcription correction algorithm has two basic phases. The first phase is training phase which is responsible to find parametric representation of each unit using HMM training algorithm. The toplogy of the HMM structure used in this experiment uses flat start to initialize the phoneme level HMM and two guassian mixture per state of the model. The second phase also called forced alignment, generates the corrected-transcription and label for each transcription and acoustic signal.

39 different feature variables are used to train and search for optimal state sequence. These are 13 Mel Frequency Cepstral Coefficient (MFCC), 13 delta coefficients and 13 acceleration coefficient from each frame. Frame size of 25 ms and frame shift of 10 ms is used.

4. EXPERIMENTAL RESULT

The transcription correction techniques has two primary outputs that can be used for various further applications. The first output is corrected-transcription of each of the transcription sentence. The second output is the speech segment labels for the acoustically motivated /pronunciation units in the corrected-transcription.

4.1. Letter to sound rules

Current research uses rule based letter to sound rule [3] which requires linguistic knowledge, and supervised way of learning in which pronunciation dictionary is readily available [11]. In this research, the pronunciation dictionary is automatically built through text transcription correction algorithm. Experiment has been conducted on Amharic, Hindi and Tamil languages using single speaker speech corpus.

The parallel word to pronunciation dictionary generated without any human intervention (unsupervised) as a result of transcription correction can be used to model a generalized grapheme to phoneme conversion algorithm. The algorithm can be seen as an incremental algorithm since any time a new transcription correction is made, its output dictionary can be added into the system so that the performance of the system will improve through time.

In this research, one of the popular machine learning algorithm (decision tree learning technique) is used to analyze the pattern how word transcription is converted into linguistically motivated pronunciation. The word in the original transcription are used to extract input features and the corresponding pronunciation can be used to identify output for input features. The LTS rule stated here is unsupervised and can be integrated to various speech processing applications.

For each instances of grapheme unit in the original transcription, the input vector consists of two of its previous grapheme, the current grapheme and two of its succeeding grapheme. The output pronunciation corresponding to the current grapheme will be given as output for the given input vector. Depending on the situation the output can be one or more grapheme (or its indexed variants) or null for deletion (no sound unit). The input and the output pairs comprises of the training feature set.

The total number of training sample set for Amharic, Hindi and Tamil are 48421, 55038 and 178287 respectively. Each training sample is a vector consists of the input and the corresponding output. The training data is given to wagon classification and regression tree (CART) to construct the optimal decision tree.

Five different types of test data sets are considered to evaluate the performance of the letter to sound rule. Each test set consists of input-output pair. The input is used to prediction and the output is used for analysis of the prediction result. The first type of test data set (TYPE_A) is the training data set itself. This tells us how the rules in the training set is consistent to each other which can also be affected by multiple pronunciation. The second type of test data set (TYPE_B) is held out data kept isolated while generating the input output pairs from the parallel corpus. This also gives a clue on the spectral consistency of the transcription correction algorithm to be used for such application as well as how good the training data is in terms of coverage.

The third type of test data set (TYPE_C) is test data set prepared from manually labeled speech data. We have a separate speech corpus for Amharic designed for speech synthesis purpose which is totally different from the speech we have used for transcription correction. This speech is manually labeled and corrected. Using the original transcription and the manually corrected speech labels, we have generated 30,914 input-output testing data pairs. The forth type of test data (TYPE_D) is test data set which is hand crafted schwa cases for Hindi language. In Hindi language, the most important issue during letter to sound conversion is finding the schwa to be deleted. From manually prepared word list and expert decision on the schwa of the words, 695 input-output pairs of test data is prepared and used for this testing purpose. The last type (TYPE_E) is also a schwa case filtered from the held out data for Hindi.

Table 1, 2 and 3 shows the performance of the decision tree learning algorithm for Amharic, Tamil and Hindi respectively. The rows shows performance at different threshold values on the minimum number of units per cluster at leaf node of the tree where as columns shows performance on different types of test data.

The general performance of the unsupervised letter to sound rule generated automatically shows high performance for test data which are prepared manually or automatically with the help of the acoustic reference as shown in table 1, 2 and 3 under column type A, B, C and E for all the three languages (Hindi, Tamil and Amharic). Tests conducted on Hindi language using test data prepared manually with language expert knowledge on only schwa case shows less performance compared to other test sets as shown in table 3 column TYPE_D. Comparing TYPE_D and TYPE_E of table 3, the minimum and maximum performance reduction are 9.978 and 11.944 respectively.

Table 1. Analysis of Amharic LTS rule performance

Stop value	Performance		
	TYPE_A	TYPE_B	TYPE_C
1	97.416	96.03	94.65
2	96.507	96.11	94.342
3	96.271	95.83	94.3
4	96.059	94.91	94.203
5	95.931	94.43	94.099
6	95.829	94.42	94.070
20	95.478	94.05	93.960

Table 2. Analysis of Tamil LTS rule performance

Stop value	Performance	
	TYPE_A	TYPE_B
1	98.202	98.26
2	98.073	98.180
3	98.023	98.020
4	97.962	97.960
5	97.935	97.940
6	97.905	97.940
20	97.604	97.600

Table 3. Analysis of Hindi LTS rule performance

Stop value	Performance			
	TYPE_A	TYPE_B	TYPE_D	TYPE_E
1	99.064	98.73	85.528	97.472
2	98.799	98.55	85.528	96.067
3	98.686	98.450	85.528	95.787
4	98.577	98.450	86.397	95.225
5	98.526	98.450	85.528	95.506
6	98.443	98.350	84.949	94.944
20	97.851	97.950	84.515	93.540

This variation is attributed to partly to the expert that label the manual data, partly to the transcription correction algorithm which may fail to get the acoustic units about the sound unit at a particular segment and partly the classification and regression algorithm failure to generalize from the given rules input-output pairs.

The performance difference between TYPE_A and TYPE_B in table 2 is insignificant and shows the performance convergence property. The convergence is mainly due to the huge amount of training data prepared for Tamil. The small variation is also attributed to the random selection nature of wagon in decision making process.

5. CONCLUSION

One of the most important feature required for improvement of grapheme based speech systems is pronunciation dictionary. In this paper, we propose an approach to extract vital linguistic information to build and improve speech synthesis and recognition systems through transcription correction. One of the output of the transcription correction algorithm, pronunciation transcription, is used to build unsupervised letter to sound rule. The performance of the letter to sound rule has been tested for three language Amharic, Hindi and Tamil on different test data types.

In this paper we tried to show techniques to find spectrally consistent grapheme to phoneme mapping with the help of the acoustic data. The same approach can be used to map the intended pronunciation of a word to the pronunciation uttered by the speaker for that word. It has been shown that, the unsupervised letter to sound rule generated automatically with the help of the transcription correction algorithm perform well on test data prepared on the basis of spectral information for all test data and languages.

6. REFERENCES

[1] Mirjam Killer, Sebastian Stuker, and Tanja Schultz, “Grapheme based speech recognition,” *Proceeding of Eurospeech*, 2003.

[2] Sebastian Stiiker and Tanja Schultz, “A grapheme based speech recognition for russia,” *Speech and Computer (SPECOM)*, 2004.

[3] Alan W. Black, Kevin Lenzo, and Vincent Pagel, “Issues in building general letter to sound rules,” *3rd ESCA Workshop on Speech Synthesis*, 1998.

[4] Simo Broman and Mikko Kurimo, *Methods for combining Language models*, INTERSPEECH 2005, Lisbon, Portugal, 2005.

[5] Alan W. Black and Ariadna Font Llitjo’s, “Unit selection without a phoneme set,” *IEEE TTS Workshop*, 2002.

[6] R Sproat, M. Ostendorf, and Andrew Hunt. (Eds.), “The need for increased speech synthesis research: Report of the 1998 nsf workshop for discussing research priorities and evaluation strategies in speech synthesis.,” Tech. Rep., 1999.

[7] Paisarn Charoenpornsawat, Sanjika Hewavitharana, and Tanja Schultz, “Thai grapheme-based speech recognition,” *Proceedings of Human Language Technology Conference of NAACL*, 2006.

[8] S. Kanthak and H. Ney, “Context dependent acoustic modeling using grapheme for large vocabulary speech recognition,” *Proceedings of ICASSP*, 2002.

[9] S. Kanthak and H. Ney, “Multilingual acoustic modeling using grapheme,” *Proceedings of European Conference on speech communication and technology*, 2003.

[10] Sebsibe Hailemariam, S.P. Kishore, Rohit Kumar, Alan Black, and Rajeev Sangal, “Unit selection voice for amharic using festvox,” *ISCA*, 2004.

[11] Vincent Pagel, Kevin Lenzo, and Alan W. Black, “Letter to sound rules for accented lexicon compression,” *ICSLP98, vol 5*, vol. 2015-2020, 1998.