Context Dependent Statistical Augmentation of Persian Transcripts

Panayiotis G. Georgiou, Hooman Shirani Mehr, and Shrikanth S. Narayanan

Speech Analysis and Interpretation Laboratory, http://sail.usc.edu
Department of Electrical Engineering and Integrated Media Systems Center
University of Southern California, Los Angeles, CA 90089, USA
[georgiou,shri]@sipi.usc.edu, shiranim@usc.edu

Abstract

Persian language is transcribed in a lossy manner as it does not, as a rule, encode vowel information. This renders the use of the written script suboptimal for language models for speech applications or for statistical machine translation. It also causes the text-to-speech synthesis from a Persian script input to be a one-to-many operation.

In our previous work, we introduced an augmented transcription scheme that eliminates the ambiguity present in the Arabic script. In this paper, we propose a method of generating the augmented transcription from the Arabic script by statistically decoding through all possibilities and choosing the maximum likelihood solution. We demonstrate that even with a small amount of initial bootstrap data, we can achieve a decoding precision of about 93% with no human intervention. The precision can be as high as 99.2% in a semi-automated mode where low confidence decisions are marked for human processing.

1. Introduction

1.1. The Transonics Project

The Transonics *Speech-to-Speech* (S2S) [1] translation system, developed as part of the DARPA speech translation program [2], aims to enable a doctor-patient interaction in the spoken languages of English and Persian respectively. The system comprises of several sub-components that act in a collaborative manner, and a visualization and control graphical user interface (GUI).

A functional block diagram, shown in Fig. 1, demonstrates the individual subsystems: two class-based *Automatic Speech Recognition* (ASR) systems (currently 23k words in English and 9k words in Persian); a *Dialog Manager* (DM), which controls the dialog flow according to the history of the conversation and is responsible for user confirmation and user choice selection; a *Machine Translation* (MT) unit, which works in a classifier mode for a limited set of utterances, or in the case of poor classification confidence, in a fully stochastic manner; and a *Text To Speech* synthesizer (TTS) that operates in a unit selection mode with a back-off to a diphone synthesizer.

This project requires the development of the aforementioned components in both English and Persian.

1.2. The Persian Language

Written Persian employs the Arabic script, an orthographic representation that does not contain explicit vowel transcriptions. This creates an ambiguity that renders written Persian text suboptimal for use in speech to speech applications. Similar complications exist in the Arabic language,[1] for which some work has been presented in [3, 4].

In a Speech-to-Speech system, two main complications arise due to the lack of vowel transcription in the Persian language:

The first one concerns the Language Model (LM), where many states are wrongly merged, thus unnecessarily biasing our recognition performance.

Second, the *Text-To-Speech* synthesis (TTS) has insufficient information to correctly synthesize speech. A one text-to-many sounds mapping results in wrong pronunciation output, thus detrimentally influencing human understanding of the synthesized speech. Common *Letter To Sound* (LTS) rules, such as those employed by the Festival system [5], are inadequate to recover the input signal's missing information.

An added benefit stemming from the ability to correctly identify pronunciations is the generation of dictionaries [3], although current accuracy measures for completely unseen words make this, in our opinion, a secondary goal.

In our recent work, and in order to eliminate the homograph ambiguity arising from the vowel absence in the Persian language, a transcription system that encodes additional vowel information was proposed. The work by Ganjavi *et al* [6], presents three transcription schemes:

USCPers: This Romanized scheme has a 1-1 mapping with the Arabic script

USCPron: A Romanized, IPA-based phonetic transcription of the Persian language

USCPers+: Combining information from both USCPers and USCPron (*i.e.,* the orthographic and phonetic alphabets) into the USCPers+ transcription script results in a scheme that avoids both the homograph and homophone issues

Since USCPers+ also includes pronunciation information, the conversion of the Arabic script to the USCPers+ method addresses both of the aforementioned complications.

The drawback of augmented transcription schemes is the lack of existing data in that format. In this paper, we propose to address this issue by employing a semi-automated transliteration process. We will attempt to identify and correctly transliterate the words for which the system gives high confidence scores, and mark as out-of-coverage low confidence decisions. In the case of off-line data collection, our goal is to minimize wrong decisions at the cost of labeling more data as out-of-coverage, while in the case of real-time TTS from non-augmented script

[1] Arabic vowel information is sometimes represented as diacritics in religious or children's texts. In Persian, although the same transcription rules exist, we have not observed any transcripts employing the accented characters.

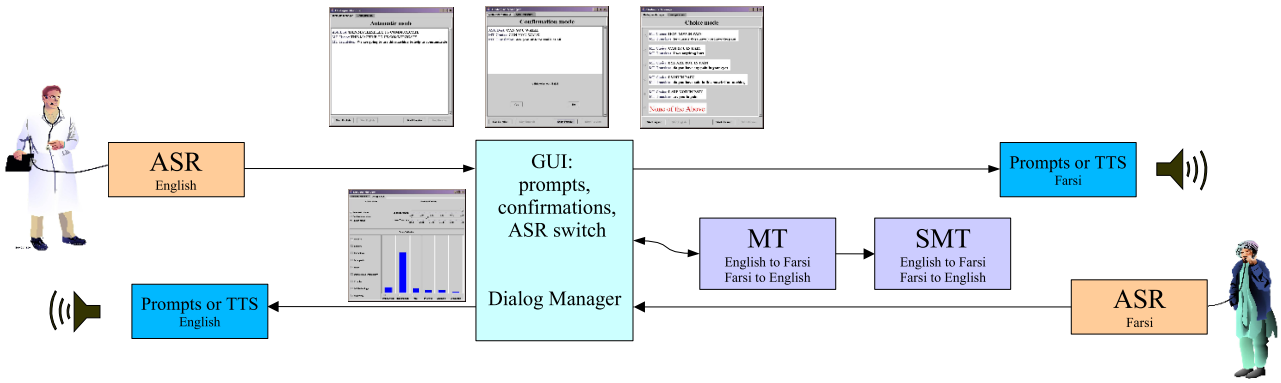

Figure 1: Block diagram of system. The output of the Persian recognizer needs to encode as much content of the ASR input as possible. The current transcription scheme of the Persian language is lossy, and thus suboptimal for use in this task.

text, the task will be to provide complete coverage even at the cost of lower precision.

2. Information recovery methods

The lack of vowels in the transcription scheme suggests a lossy encoding of the underlying language. The information recovery process that humans employ in reading Persian transcripts can be categorized in three distinct, yet highly related techniques:

2.1. C1: One-to-one mapping

The consonant transcription uniquely identifies the word: 1-1 mapping between the USCPers – USCPers+ schemes. A unigram of occurrences of the language can help completely recover the lost information as demonstrated in the "mn" example of Fig. 2.

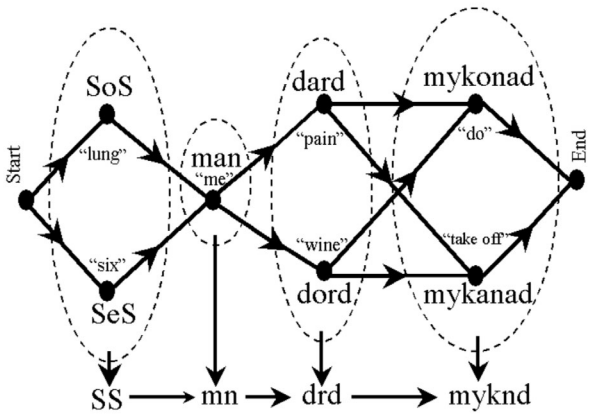

Figure 2: Decoding through hidden states – all possible USCPers+ variants of the observed USCPers word – in order to estimate the Maximum Likelihood hypothesis. Diagram only shows the valid USCPers+ alternatives, *i.e.,* what has already been observed through the training phase.

2.2. C2: Short term context

The preceding and following words are sufficient to identify the concept, and hence the pronunciation. In this case, language model decoding can aid in recovering the lost information as demonstrated in Fig. 2. The path "SoS man dard mykonad" is the maximum likelihood one in this case, since training data have resulted in a language model of low probability for the alternative n-grams.

However, this procedure will be lossy, and the degree of degradation will significantly depend on the coverage of the USCPers+ language model.

2.3. C3: Long term context or discourse

The surrounding words are insufficient for augmentation as shown in Fig. 3, but the long term discourse or topic is defining the meaning. In such a case, the order of the required language model will be large, and hence impractical. However, we expect that the occurrences of this third category are few.

The above augmentation can be treated as a maximum likelihood path estimation by considering the augmented transcription as the hidden states, and the USCPers version as the observed states. Fig. 2 demonstrates the concept with the decoding of the phrase "I have pain in my lungs" (gloss: "lung me pain do").

3. Objective and proposed solution

The objective is to maximize the available data in the USCPers+ transcription scheme for training LMs for speech recognition

Arabic script	این کلفت است.	
USCPers	Ayn klft Ast	
Gloss	this thick is	this maid is
Translation	This is thick	This is a maid
USCPers+	Ayn koloft Ast	Ayn kolfat Ast

Figure 3: An example in which the short term (bigram in this case) information is insufficient for ambiguity resolution.

and machine translation technologies. A second objective is to generate pronunciations for a TTS synthesizer in the absence of an augmented transcription input, noting that letter to sound rules are not sufficient in the Persian language.

Solution 1: Given a transcribed sequence $\underline{w^a} = [w_1^a w_2^a \dots w_n^a]$ in the Arabic script, convert it into the equivalent (lossless conversion) Romanized version $\underline{w^r} = [w_1^r w_2^r \dots w_n^r]$ (USCPers script). Subsequently, generate all the possible USCPers+ transcriptions u_i^r, and choose the maximum likelihood USCPers+ sequence $\underline{\hat{u}^r}$ based on prior training data.

The maximum likelihood solution can be expressed through the decoding of a Markov process [7]. The requirements for implementing the above method is bootstrap training data for generating the language model in the USCPers+ domain. While transliteration is taking place, the generated data can be used to augment the USCPers+ LM. Additionally, it is desired – although not necessary – to have a manually generated mapping between the USCPers and USCPers+ languages. This aids in improving the unigram counts of the language model by disabling invalid words and making sure that in a 1-to-N mapping scenario, all N USCPers+ pairs appear in the LM. Nevertheless, in the absence of a mapping table, all USCPers+ possibilities can be generated; there are 4 valid options for the inter-consonant positions in a word: insertion of a vowel ("a", "e", or "o") or no insertion.

4. Results

As described earlier, we can identify two distinct usages for augmented transcriptions, and hence two distinct optimization criteria:

1. For the offline processing of data and with the purpose of augmenting the training corpus, we would like to maximize the precision:

$$\text{precision} = \frac{\text{words correctly transliterated}}{\text{total words of input}} \tag{1}$$

2. For the real-time augmentation of the transcription scheme for a TTS synthesizer, we would like to eliminate human intervention. We define the measure of *coverage* as:

$$\begin{aligned}\text{coverage} &= \frac{\text{total words transliterated}}{\text{total words of input}} \\ &= 1 - \frac{\text{\# of "no decision can be made" words}}{\text{total words of input}}\end{aligned} \tag{2}$$

Eliminating such intervention without loss in precision would require very good coverage from the existing language models. In the absence of sufficient LM coverage, shortcuts that will undoubtedly decrease precision can be taken, such as predicting the vowels at the letter level. This statistical prediction approach to Letter-To-Sound rules will be discussed at the end of this paper.

The raw database used for our experiments is a corpus of over 32k words, extracted from publicly available sources such as the Hamshahri newspaper on-line archive. The Arabic script is then automatically transformed to the Romanized scheme, USCPers, and converted to the augmented USCPers+ scheme by human experts. The resulting USCPers+ data is utilized for building a language model using the SRILM toolkit [8]. In order to test the system, we used the "leave-one-out" technique and partitioned the corpus into different test and training subsets for cross-validation. Table 1 demonstrates the experimental results under different data partitioning conditions.

Words	Train	Test	Coverage	Precision
Test 1	16155	16155	61.1%	98.8%
Test 2	19262	13048	70.7%	98.6%
Test 3	22625	9685	80.7%	98.3%
Test 4	25855	6455	83.6%	98.7%
Test 5	29079	3231	90.6%	98.8%

Table 1: Results of transliteration precision and coverage with no rejection metrics in place – Solution 1.

As can be observed from the above results, the performance is not improving as the amount of data increases. In such a data starved scenario, an increase in the training data results in an increase in the vocabulary size, thus underscoring the sparsity effects of our language model. For example, even after the training data vocabulary size has reached about 5800 unique words, an additional 5% of training data introduces about 200 more new words. Assuming a trigram model, an order 3 operation, the desired corpus size growth under these conditions would be of the order of 10%. Additionally, in the case that the data are recursively collected, augmented, and added to the training corpus, there would be an accumulation of errors detrimental to the convergence of the LM. In such a case, much higher decision is desired.

5. Higher Precision, Higher Rejection

To counter for the above sparsity, we propose a variation to the aforementioned method. The suggested solution is based on confirming whether the ML choice was an informed decision made in light of sufficient data, or whether the decoding was performed in a sparse coverage area.

Solution 2: Proceed as in Solution 1 above and derive a decision as to the correct transliteration. After the maximum likelihood decision is made, determine its viability based on a thresholding operation.

The suggested thresholds are the following, with Level 1 being the strictest (higher precision):

L1 Decision was made based on 3 trigrams

L2 Decision was made based on at least 2 trigrams

L3 Decision was made based on at least 1 trigram

L4 Decision was made based on at least 2 bigrams

L5 Decision was made based on at least 1 bigram

L6 Decision was made based on at least 1 unigram (equivalent to no thresholding, as in the previously proposed method)

We define here *thresholded coverage* as

$$\text{T. coverage} = \frac{\text{\# of words with accepted transliteration}}{\text{total words of input}} \tag{3}$$

As shown in Table 2, the precision increases at L1 with data growth, while from the last row, L6, no significant improvement in precision can be observed despite the augmented training corpus size. A result that needs to be investigated with a much larger corpus is the variability among levels 1-3.

Thr. Levels	Test Conditions				
	T1	T2	T3	T4	T5
L 1	98.9%	98.6%	98.6%	99.0%	99.2%
L 2	98.9%	98.6%	98.6%	99.0%	99.2%
L 3	98.9%	98.6%	98.6%	99.0%	99.2%
L 4	98.8%	98.6%	98.3%	98.7%	98.8%
L 5	98.8%	98.6%	98.3%	98.7%	98.8%
L 6	98.8%	98.6%	98.3%	98.7%	98.8%

Table 2: Precision under Thresholding

Thr. Levels	Test Conditions				
	T1	T2	T3	T4	T5
L 1	60.4%	68.4%	75.4%	73.9%	77.1%
L 2	60.4%	68.4%	75.4%	74.4%	78.6%
L 3	60.4%	68.4%	76.2%	76.9%	84.0%
L 4	61.1%	70.6%	80.7%	83.6%	90.6%
L 5	61.1%	70.7%	80.7%	83.6%	90.6%
L 6	61.1%	70.7%	80.7%	83.6%	90.6%

Table 3: Thresholded Coverage

Similarly, in Table 3, levels 1-3, we can see a significant decrease in the thresholded coverage, while the precision at these levels remains constant within our test-set. This suggests that a L3 may be sufficiently restrictive to detect and reject most of the decoding errors. This also needs to be investigated with a larger data set.

6. Unseen words: Statistical LTS

In contrast to increasing precision, 100% coverage is required in pronunciation rendering from textual information. However, due to the amount of new words introduced with new data, even the unigram decision of L6 is not sufficient to provide complete coverage. To force complete coverage, one can generate all possible USCPers+ variations of the input USCPers word, and statistically select the most likely one.

In Persian, for each possible consonant pair, 3 different vowels may occupy the space in between: “a”, “e”, “o”. It is also possible that the consonant sequence occurs with no intermediate vowels. This results in 4^N variations for each N-letter USCPers word. We propose employing a *Statistical Letter To Sound* (SLTS) model through the use of a trigram Markov chain from a corpus of existing data.

Table 4 shows poor performance as baseline when the word is chosen among all possible ones without any priors. In the case of a Statistical letter-to-sound rule-set however, although the precision is still low in absolute terms, we observe a significant improvement.

Clearly, this is insufficient for applications such as dictionary generation, or off-line LM data corpus augmentation. It is however necessary in the TTS paradigm for the approximately 10% of out-of-coverage words. As an example, in the case of Test 5 conditions with L4 thresholding, the precision after application of the SLTS method would be: $90.6\% \times 98.8\% + 9.4\% \times 43\% = 93.6\%$, but with 100% coverage.

7. Conclusion

We proposed a semi-automated transliteration scheme that can recover information lost in the orthographic transcription of the Persian language. The system is based on decoding through the possible augmented representations, and can achieve an accuracy of over 98% in supervised operation, and over 90% when unsupervised. Additionally, by introducing a confidence threshold, the error in transliteration has been shown to decrease with the training data increase.

The augmentation of the Persian transcription scheme is necessary for TTS applications where a one-to-many mapping can not be dealt with at the synthesizer. Additionally, it can be used to pre-process training data to modify the internal representation of language models, and as such, reduce language model smoothing and bias towards merged states.

	Statistical LTS	Baseline
Test 1	41%	17%
Test 2	47%	21%
Test 3	45%	19%
Test 4	50%	25%
Test 5	43%	23%

Table 4: Vowel prediction for out-of-coverage words

8. References

[1] S. Narayanan, P. G. Georgiou, et al., “Transonics: A speech to speech system for English-Persian interactions,” in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU*, 2003.

[2] DARPA, “Information technology programs office,” `http://www.darpa.mil/ipto/programs/cast/`.

[3] K. Kirchhoff, J. Bilmes, S. Das, N. Duta, M. Egan, G. Ji, F. He, J. Henderson, D. Liu, M. Noamany, P. Schone, R. Schwartz, and D. Vergyri, “Novel approaches to Arabic speech recognition: Report from the 2002 Johns-Hopkins Workshop,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, number 1, pp. 344–7.

[4] Laura Mayfield Tomokiyo, Alan W Black, and Kevin A. Lenzo, “Arabic in my hand: Small-footprint synthesis of Egyptian Arabic,” in *Eurospeech*, 2003.

[5] “The Festival Speech Synthesis System,” `http://www.cstr.ed.ac.uk/projects/festival/`.

[6] Shadi Ganjavi, Panayiotis G. Georgiou, and Shrikanth Narayanan, “ASCII based transcription schemes for languages with the Arabic script: The case of Persian,” in *ASRU*, 2003.

[7] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall PTR, 2001.

[8] A. Stolcke, “SRILM – an extensible language modeling toolkit,” in *Proc. Intl. Conf. on Spoken Language Processing*, 2002, vol. 2, pp. 901–904.