Statistical Vowelization of Arabic Text for Speech Synthesis in Speech-to-Speech Translation Systems

Liang Gu, Wei Zhang, Lazkin Tahir and Yuqing Gao
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

ABSTRACT
Vowelization presents a principle difficulty in building text-to-speech synthesizers for speech-to-speech translation systems. In this paper, a novel log-linear modeling method is proposed that takes into account vowel and diacritical information at both the word level and character level. A unique syllable based normalization algorithm is then introduced to enhance both word coverage and data consistency. A recursive data generation and model training scheme is further devised to jointly optimize speech synthesizers and vowelizers for an English-Arabic speech translation system. The diacritization error rate is reduced by over 50% in vowelization experiments.

1. INTRODUCTION
The developments and applications of Speech-to-Speech Translation (SST) technology are growing rapidly. Most modern SST systems consist of a cascaded set of components including automatic speech recognition, spoken language translation, and Text-To-Speech synthesis (TTS). Among them, TTS is the last but not the least important component, which synthesizes speech corresponding to the translated text in the target language. Moreover, TTS is perhaps one of the most important techniques for some SST applications where text based information is not accessible, such as eyes-free hands-free speech translation and call-center based automatic translation services.

Current SST systems are usually developed for specific language pairs in some specific domains such as medical care and travel. Designing a TTS component for such SST systems requires significant investigation into the language or linguistics of a given region and a given domain. As a result, the challenges of TTS may vary among different languages. In an English-Arabic SST system, for example, the design of English TTS is to some extent easier than the design of Arabic TTS, because English vowels always appear in written words whereas in Arabic short vowels are omitted in the translation output. This is usually referred as a diacritization or vowelization problem. Therefore, the first and most important step in designing an Arabic synthesizer is to restore the vowels of each word in the translated text. In this paper, we concentrate on designing a vowelizer of Arabic text for speech synthesis in a colloquial English-Arabic speech-to-speech translation system for limited domains in emergency medical care.

There are two popular ways in which the missing vowels can be restored. The first and most common method is to assign a set of possible vowelizations to each Arabic word and store these in a vowelization lexicon, which is usually referred as the dictionary method. The vowelization lexicon may be further enumerated to improve word coverage [1]. However, even the enumerated lexicon may not cover all the words in a specific SST domain. Moreover, for colloquial Arabic speech, the accuracy of an enumerative vowelization lexicon is doubtful.

A common approach to mitigate this problem is to compliment this dictionary approach with the other vowelization approaches based on automatic generation.

The generation approaches as the other vowelization methodology infer missing vowels either by rules or based on probability models designed for the language and the domain. For the former approach, significant amount of linguistic knowledge is required that may not be available for some colloquial or dialectal Arabic languages. For the latter approach, significant amount of training data needs to be collected that is representative of both the language and the domain. In addition, the data used for training the TTS models and vowelization models should be consistent. Statistical vowelization methods proposed so far include maximum-entropy modeling [2], weighted finite transducer [3], hierarchical N-gram modeling [4], Neural Network [5], etc. While some successes have been reported on automatic vowelization for Arabic speech recognition, the vowelization accuracy still needs to be improved for high-quality Arabic TTS.

In particular, three challenges remain open in the vowelization of Arabic text for speech synthesis in SST systems. First, the vowel generation accuracy for unknown words needs to be enhanced. Second, the vowelization training corpora created by various human annotators need to be normalized for higher consistency. Additionally, the setup procedure of vowelization training data and lexicon should be joined optimized with the setup procedure of the training data for TTS and machine translation in a SST system.

In this paper, we propose a set of new approaches to handle the above concerns. A log-linear model is introduced to predict missing vowels of unvowelized Arabic translation. The word and character based vowelization models are trained on manually vowelized sentence. For known words, a Viterbi decoder is applied to obtain best vowelization hypothesis at the word level with a back-off N-gram scheme. For unknown words, a Viterbi decoder is proposed to obtain best vowelization hypothesis at the character level.

To reduce data inconsistency, an initial vowelization model is trained on the vowelized TTS training scripts solely. Other training materials are then verified and normalized at the syllable level using the initial vowelization model. An enhanced vowelization model is then trained on both the TTS training scripts and other normalized train materials.

To jointly optimize vowelization and other components in a SST system, we devised an automatic script algorithm to achieve both full syllable coverage and word N-gram balance with merely thousands of sentences. This script is manually vowelized and used for building both TTS models and initial vowelization models. A vowelization lexicon is then created that covers the most frequently appeared words in the SST domain that are missed in the generated TTS training script.
This vowelization dictionary is further integrated with the statistical vowelization model to achieve best vowelization performance. Based on this initial model, additional vowelized sentences can be further normalized and verified. This will in turn enhance the consistency of training data and result in more accuracy and robust vowelization models.

2. VOWELIZATION OF ARABIC TEXT

A. Arabic Diacritics

Arabic writing system consists of 36 letterforms that represent the Arabic consonants or glottal stops. They are:

\( \{ \) َت ُخ ُج ُح ُد ُج ُز ُس ُش ُص ُض ُط ُع ُف ُف ُق ُل ُم ُم ُو ُي ُو ُؤ ُه ُا \( \)\}. In addition to the above letters, there are 8 symbols (or marks) called diacritics that may be placed above or below a letter: َْ،ََّ،ََِ،َََ،ََََ،َََََ. In this list, three diacritics َََ،ََََ represent three Arabic short vowels, respectively. There are three more diacritics called tanweens َََ،ََََ that occur only at the end of each utterance. ََََ doubles the duration of the previous consonant, while َََ means the previous letter is neither diacritized nor gminated.

Almost all written Arabic texts are represented using the 36 consonant letters only without any diacritic symbols. This may lead to ambiguity in meaning and pronunciations. To avoid the ambiguity, the missing diacritics must be restored, either by Arabic readers in our daily lives or by the computer for a SST system. This procedure is commonly referred as diacritization or vowelization. We will use the latter term in this paper.

For English-Arabic SST systems, almost all the Arabic training data for machine translation are unvowelized, i.e., there are no diacritics in the translated Arabic text. The synthesized speech based on this text may be either not understandable or lead to a wrong meaning, which will greatly deteriorate the intelligence of the synthesized outputs. Additionally, if concatenative TTS algorithms are used, the missing or incorrectly inserted short vowels could make the naturalness and smoothness of the synthesized sound dramatically worse. Therefore, an automatic Arabic vowelizer becomes a principle component for high-quality TTS in an English-Arabic SST system. Next, we design such vowelizers using statistical models.

B. Problem Formulation

The main task of a statistical vowelizer is to infer a vowelized word sequence from a given unvowelized word sequence. Denote the input sequence of \( N \) unvowelized words as \( \textbf{U} = \{ u_1, u_2, \ldots, u_N \} \) and the output sequence of \( N \) vowelized words as \( \textbf{V} = \{ v_1, v_2, \ldots, v_N \} \); the best vowelization hypothesis can be achieved by maximizing conditional probability as

\[
\hat{\textbf{V}} = \arg \max_{\textbf{V}} \ P(\textbf{V}|\textbf{U}) = \arg \max_{\textbf{V}} P(\textbf{v}_1, \textbf{v}_2, \ldots, \textbf{v}_N | \textbf{u}_1, \textbf{u}_2, \ldots, \textbf{u}_N) .
\]

If there are unknown words in the input sequence, a statistical vowelizer should also be able to infer a vowelized word from a given unvowelized word. Assume the input unknown word consists of \( M \) consonant letters as \( \textbf{U} = \{ C_1, C_2, \ldots, C_M \} \) and its corresponding vowelization is consisted of the same set of consonant letters and \( M \) additional diacritical marks as \( \textbf{V} = \{ c_1, d_1, c_2, d_2, \ldots, c_M, d_M \} \). Note that these \( M \) diacritical marks can be one of the 8 symbols or the combination of two of the 8 symbols listed in section 2.A. The marks could also be empty. Namely, the real variations of diacritical marks in \( \textbf{V} \) could be more than 8 and the real number of diacritical marks in \( \textbf{V} \) could be less than \( M \).

Similar to equation (1), the best vowelized-word hypothesis can be achieved as

\[
\hat{\textbf{V}} = \arg \max_{\textbf{V}} P(\textbf{v}_1, \textbf{v}_2, \ldots, \textbf{v}_N | \textbf{u}_1, \textbf{u}_2, \ldots, \textbf{u}_N) .
\]

Vowelization training data is usually limited, especially for the colloquial and dialectal languages. Therefore equations (1) and (2) need to be significantly simplified, as we propose next.

C. Log-Linear Modeling

The conditional probabilities in equation (1) can be written as

\[
P(\textbf{v}_1, \textbf{v}_2, \ldots, \textbf{v}_N | \textbf{u}_1, \textbf{u}_2, \ldots, \textbf{u}_N) = \prod_{n=1}^{N} P(v_n | u_n, u_{n-1}, u_{n-2}, \ldots, u_1) .
\]

where \( T(\textbf{u}, \textbf{v}) \) is a test function that checks whether word \( \textbf{v} \) is a valid vowelization of word \( \textbf{u} \).

\[
P(\textbf{u}_1, \textbf{u}_2, \ldots, \textbf{u}_N) = \beta_n \text{ is a constant for each input sequence.}
\]

Hence, equation (1) can be written as

\[
\hat{\textbf{V}} = \arg \max_{\textbf{V} \in \textbf{U}^N} \left\{ \sum_{n=1}^{N} \log P(v_n | u_n, u_{n-1}, u_{n-2}, \ldots, u_1) \right\} .
\]

The first item in equation (4) represents the conditional probability of a vowelized word hypothesis give the input sequence, while the second item in equation (4) indicates the relationship among generated vowelized words given the input sequence. These two items can be trained from the frequency of occurrence of parallel vowelized/unvowelized word sequences in the vowelization training data.

Because of the limit amount of vowelization training data available for most SST applications, equation (4) may be modified using log-linear models that are well known in the machine translation area [6]:

\[
\hat{\textbf{V}} = \arg \max_{\textbf{V} \in \textbf{U}^N} \left\{ \alpha \sum_{n=1}^{N} \log P(v_n | u_n, u_{n-1}, u_{n-2}, \ldots, u_1) + \right\}
\]

1902
where $\alpha$ is the word-based log-linear probability weight that can be tuned on a held-out development set.

Equation (2) can be modified in the same manner and written in the log-linear form as

$$
\hat{V} = \arg \max_{V \in U} \beta \sum_{n=1}^{N} \log P(d_n | v_1, \ldots, v_n) + (1 - \beta) \sum_{n=2}^{N} \log P(d_{n-1}, d_n | d_{n-2}, c_n, \ldots, c_0),
$$

(6)

where $\beta$ is the character-based log-linear probability weight that can be tuned on a held-out development set.

\[D. \text{Hierarchical Viterbi Search}\]

To statistically restore diacritical information, Viterbi search can be performed based on either equations (5) at the word level or equation (6) at the character level. A better approach is to apply Viterbi decoding hierarchically, as what was proposed in [4]. For a given unvowelized sequence $U = \{u_1, u_2, \ldots, u_N\}$, the best vowelization hypothesis $\hat{V} = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_N\}$ is achieved as follows:

1) Get unknown word list $W = \{u_j | 1 \leq j \leq N \leq L \leq L \leq U\}$.
2) If $W = \phi$, then set $\hat{X} = \phi$ and go to step 3). Otherwise, obtain $\hat{X} = \{\hat{v}_j | 1 \leq j \leq N \leq L \leq L \}$ using Viterbi search based on equation (6).

3) Use Viterbi search to achieve best vowelization hypothesis $V$ as

$$
\hat{V} = \arg \max_{V \in U} \left\{ \sum_{n=1}^{N} \log P(v_1, \ldots, v_n) + (1 - \beta) \sum_{n=2}^{N} \log P(v_{n-1}, v_n | d_{n-1}, \ldots, d_0) \right\},
$$

(7)

A back-off scheme is further applied to the use of estimated conditional probabilities in equation (6) (7), similar to what is commonly used in N-gram based language modeling for speech recognition [7]. For any $m$ and $n$, we make back-off approximations of conditional probabilities as

$$
P(d_{n-1}, \ldots, d_0 | d_n, v_1, \ldots, v_n) = P(d_n | v_{n-1}, \ldots, v_1) \cdot P(d_{n-1}, \ldots, d_0 | d_n, v_{n-1}, \ldots, v_1),
$$

(8)

$$
P(d_{n-1}, \ldots, d_0 | v_1, \ldots, v_n) = P(d_n | v_{n-1}, \ldots, v_1) \cdot P(d_{n-1}, \ldots, d_0 | d_n, v_{n-1}, \ldots, v_1),
$$

(9)

$$
P(v_1, \ldots, v_n | d_{n-1}, \ldots, d_0) = P(v_1, \ldots, v_n | d_{n-1}, \ldots, d_0, v_{n-1}, \ldots, v_1),
$$

(10)

$$
P(v_1, \ldots, v_n | v_1, \ldots, v_n) = P(v_1, \ldots, v_n | v_1, \ldots, v_n, d_{n-1}, \ldots, d_0),
$$

(11)

3. ENHANCE VOWELIZATION CONSISTENCY VIA SYLLABLE BASED NORMALIZATION

A. Problem of Vowelization Inconsistency

While vowelization defines the sense of each Arabic word and its corresponding pronunciations, the definition of the vowelization of each word varies very significantly. For a given unvowelized word $u = \{c_1, c_2, \ldots, c_M\}$, there are usually a set of corresponding vowelizations with $L$ kinds of written forms as $v = \{v_1, v_2, \ldots, v_L\}$. For Arabic readers, all these $L$ ways of vowelization are correct and understandable.

For a SST system, however, it will be a disaster if the definition adopted for TTS models is inconsistent with the definition adopted for vowelization models. The resulting synthesized sound will be terrible because some short vowels and diacritics ignored in vowelization may be critical for a TTS algorithm, while some short vowels and diacritics generated via automatic vowelization may be totally unknown for TTS models and hence leads to unpredictable synthesized sounds.

In practice, however, the vowelized text utilized in TTS training is often the transcription of the recorded speech and hence always very limited. On the other hand, there are usually much larger amount of vowelized texts derived from other resources and annotated by additional Arabic speakers. Therefore, it is an open challenge of using the additional but inconsistent vowelization data to enhance TTS quality in SST systems.

B. Syllable-based Normalization of Vowelized Words

To tackle the above challenge, we propose a novel approach that performs syllable-based normalization on additional vowelized texts other than TTS training scripts.

Denote $c = \{c_{j,1} | 1 \leq j \leq 36\}$ and $d = \{d_{j,1} | 1 \leq j \leq J\}$ are the Arabic consonant letter set and diacritic set, respectively. Define Arabic syllable set as $s = \{c d c’ | c, d, c’ \in c\}$. Assume there are $K_s$ vowelized words $\{v_k, 1 \leq k \leq K_s\}$ in the TTS training scripts and $K_a$ vowelized words $\{v_k, 1 \leq k \leq K_a\}$ in the additional training materials, our proposed normalization scheme performs the follow steps:

1) Collect all the syllables occurred in $\{V_k\}$ and build TTS syllable set $s = \{c d c’ | c, d, c’ \in c, c d c’ \in c\}$.
2) Train vowelization model on $\{V_k\}$ using equation (5,6);
3) For each word $V_k$, in the additional training data, get the corresponding unvowelized word $U_k$ and infer its vowelization $\tilde{V}_k$ via models built in step 2);
4) For each syllable $c d c’$ in $\tilde{V}_k$, if $c d c’ \notin s$, then normalize $c d c’$ to $\tilde{V}_k$ in $\tilde{V}_k$; Hence get normalized word $\tilde{V}_k$;
5) Perform steps 3) and 4) until all the words in $\{V_k\}$ are normalized as $\tilde{V}_k$; Rebuild vowelization model using $\{V_k\}$ and $\tilde{V}_k$.

4. DESIGN STATISTICAL VOWELIZERS FOR TTS IN SST SYSTEMS

To optimize Arabic vowelizers in SST systems, we propose a new design methodology that builds vowelization models and
Figure 1. Flow-chart of designing statistical vowelization models for TTS in a SST system

TTS models simultaneously. The flow chart of data preparation and model training is depicted in Figure 1. For a given set of bilingual parallel unvowelized set $P_{mt}$ that represents the domain of a SST system, a set of sentences $U_{tr}$ are generated via an automatic script algorithm to achieve both full syllable coverage and word N-gram balance with merely thousands of sentences. This set is then manually vowelized into a vowelized set $V_{TTS}$. $V_{TTS}$ is used as the scripts for recording speech waveforms and training TTS models. In the meantime, a vowelization lexicon $L_m$ is built that covers the words that occur most frequently in $P_{mt}$ but never appear in $V_{TTS}$. The initial vowelization model $M_1$ is built upon $V_{TTS}$ and $L_m$ following the descriptions in section 2.

If there is additional vowelization training set $V_A$, it needs to be normalized into $\tilde{V_A}$ as described in section 3. The vowelization model is thereafter enhanced as $\tilde{M}_V$ using training sets $V_{TTS}$ and $\tilde{V_A}$ as well as the vowelization lexicon $L_m$. The new model may then be utilized to re-normalize the additional data $\tilde{V_A}$. This procedure is iterated until the model parameters in $\tilde{M}_V$ are converged.

5. EXPERIMENTS

The performance of our proposed vowelization algorithms and schemes were evaluated on the English-Arabic speech-to-speech translation task within a limited domain of emergency medical care. Altogether 550k colloquial in-domain sentences in both English and Arabic were collected for evaluation. About 7000 vowelized sentences were generated from the 550k in-domain sentences and cover about 3500 syllables. These sentences were used as TTS recording scripts. It is then split into training (80%), dev (10%) and test (10%) sets for vowelization experiments. About 50k vowelized sentences were used as additional materials that were prepared by different human annotators. The output of our vowelization model is about 22MB. The decoding speed is about 5000 words per second on IBM Thinkpad T42p laptops.

In the first experiment, the log-linear modeling approach was compared with conventional N-gram based approach in [4] and achieved 45% reduction in vowelization error rate as shown in Table 1. In the second experiment, the vowelization models $M_1$, $M_2$, $M_3$ were trained on $V_{TTS}$, $V_{TTS} + V_A$ and $V_{TTS} + \tilde{V_A}$, respectively (as described in section 4) and evaluated on the same test set similar to $V_{TTS}$. $M_1$ substantially outperformed $M_3$ and $M_2$ because of its superiority in both word coverage and data consistency.

We also carried out ABX test on TTS quality using both the baseline vowelization method in Table 1 and our proposed vowelization method. Preliminary results show that in 77% of time, the TTS quality using our new vowelization is better than the TTS quality using conventional vowelization methods.

6. CONCLUSION

Vowelization is a critical component in Arabic text-to-speech (TTS) synthesis for speech-to-speech translation (SST) systems. We propose a new log-linear statistical vowelization modeling method that infers short vowels and diacritics at both word and character levels. A syllable-based normalization scheme is then introduced to alleviate the discrepancy of vowelization in training data annotated by various Arabic speakers. A complete design scheme is further proposed to jointly optimize TTS models and vowelization models for a SST system. Substantial improvements were achieved in both vowelization accuracy and TTS quality.

7. REFERENCES