Abstract. This paper presents approach of multi-lingual speech corpus design, data collection and phonetic annotation for text-to-speech (TTS) system development. Under a uniform data structure, more than 10 languages and dialects speech corpora are shared with language independent data management approaches and data processing procedures. A specifically defined super phonetic symbol set are used for all languages and related dialects. The defined data management methods enable Motorola multi-lingual TTS systems employs a uniform architecture for cost function-based unit selection strategy and speech synthesizer modules on both sever-based and embedded platforms.

Keywords: Multi-lingual, TTS, speech corpus.

1 Introduction

Generally speaking, there are two approaches of text-to-speech synthesis: concatenative speech synthesis and parametric speech synthesis. Big-corpus based concatenative TTS [1] is still dominant due to its natural sounding output. For this reason, the multi-lingual TTS speech corpora target on the concatenative method. Larger speech corpus can provide more opportunities to get longer and suitable speech unit to minimize the number of splices and therefore discontinuities between contiguous units. The problem is how to get phonetic coverage on both segmental and super-segmental or prosody.

The multi-lingual TTS systems developed by Motorola China Research Center [2] cover more than 10 languages/dialects, such as: US English, UK English, European French, Canadian French, European Spanish, Am. Spanish, German, Italian, Brazilian Portuguese, Mandarin, Cantonese and etc.. These languages involve more than three language families, such as Indo-Euro family and Sino-Tibetan family.

For TTS developers are un-familiar with most of developing languages, efficient resources arrangement becomes more important. The uniform method and architecture are helpful to TTS speech corpus development.
The components of TTS systems can be classified into two categories: language independent or common modules and language dependent modules. Unit selection and speech synthesis are language independent while text processing and speech data are language dependent. For the language dependent modules, the methods and architecture are uniform. For example, script design, text normalization, letter-to-sound (L2S) conversion, phonetic symbol expression, speech unit definition and data structure of speech inventory, etc., are all uniform.

Relatively, to meet with the needs of TTS development, the major tasks of building multi-lingual speech corpus are (1) defining machine readable symbols based on phonetic systems; (2) creating TTS oriented lexicons; (3) designing script in terms of phonetic coverage; (4) selecting speakers; (5) collecting speech data; (6) phoneme segmentation; and (7) prosody annotation.

2 Phonetic systems and phonetic symbol

For multi-lingual TTS speech corpus design and generation, a super set of machine readable phonetic symbol is basis for the purpose of working efficiency. Each symbol in this set can match unique IPA symbol.

2.1 Phonetic systems

Currently, Motorola China Research Center is developing more than 10 languages multi-lingual TTS systems. These languages belong to 3 language families and 6 language groups. Reasonable description for phonetics and linguistics of these languages can reduce language complexity. Firstly, we created a super phonetic set for 12 languages (Table 2.1). The super phonetic set contains 121 phonetic identities, 48 vowels (including diphthongs) and 73 consonants (including nasals). Among these phonetic identities, only 15 phonemes are shared by all the 12 languages, while 58 phonemes are not shared by any two languages, i.e., they only appear in one language.

Table 2.1 Phonetic identities of 12 languages for TTS development.

<table>
<thead>
<tr>
<th>Language</th>
<th>ESP</th>
<th>FRE</th>
<th>CFR</th>
<th>GER</th>
<th>ITL</th>
<th>PTG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phonemes</td>
<td>33</td>
<td>39</td>
<td>48</td>
<td>48</td>
<td>44</td>
<td>43</td>
</tr>
<tr>
<td>vowels</td>
<td>5</td>
<td>15</td>
<td>22</td>
<td>23</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>Consonants</td>
<td>28</td>
<td>24</td>
<td>26</td>
<td>25</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>Language</td>
<td>RUS</td>
<td>ARB</td>
<td>POL</td>
<td>KOR</td>
<td>FIN</td>
<td>ENG</td>
</tr>
<tr>
<td>Phonemes</td>
<td>48</td>
<td>42</td>
<td>36</td>
<td>40</td>
<td>24</td>
<td>42</td>
</tr>
<tr>
<td>vowels</td>
<td>10</td>
<td>14</td>
<td>8</td>
<td>21</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Consonants</td>
<td>38</td>
<td>28</td>
<td>28</td>
<td>19</td>
<td>16</td>
<td>26</td>
</tr>
</tbody>
</table>

Note: ESP is for Spanish, FRE for French, CFR for Canadian French, GER for German, ITL for Italian, PTG for Portuguese, RUS for Russian, ARB for Arabic, POL for Polish, KOR for Korean, FIN for Finnish and ENG for English.
It should be pointed out that the super phonetic set may slightly different to that of linguistic textbook because it is for engineering realization of speech synthesis. For example, some Arabic textbook identifies 11 vowels. But based on the allophones in real pronunciation, there are 14 vowels distinctively. For instance, the Arabic phonetic identity of vowel diacritics / / has two distinctive pronunciations, / \textipa{ê} / and / \textipa{é} / (represented by IPA symbol). The three additional vowels are / \textipa{ê} /, / \textipa{ô} / and / \textipa{ê} / correlated to the phone identities / \textipa{ê} /, / \textipa{ô} / and / \textipa{ê} /.

### 2.2 Representation of Unicode and extended ASCII alphabets

Another issue in multi-lingual TTS speech corpus design is the Unicode and/or extended ASCII of text. A few of language's contains Unicode alphabets, such as Arabic and Russian, and some extended ASCII alphabets. To use a unique platform to create and employ the multi-lingual TTS speech corpus, we adopt a uniquely defined machine readable symbol for the Unicode alphabets, as well as the extended ASCII alphabets. One advantage of this approach is that it can avoid switching the regional setting of platform for languages changes. These machine readable descriptions of alphabets are referred as MCRC symbols. The rule of how to represent the Unicode and extended ASCII alphabets by MCRC symbols is that: (1) each Unicode alphabets is represented with an English alphabet, or plus a suffixal digital, e.g. e1, o2, d8, etc.; (2) the English alphabet is selected based on the pronunciation of the Unicode alphabet; (3) if an English alphabet correlates to more than one Unicode alphabets in an individual language, different valued suffixal digits are added to distinguish them. For instance, “e1” and “e2” are used to represent “ê” and “é”, respectively. Table 2.2 gives examples of the machine readable description set of the Unicode or extended ASCII alphabets.

#### Table 2.2 Examples of simple representation for unicode alphabets

<table>
<thead>
<tr>
<th>Unicode</th>
<th>ê</th>
<th>é</th>
<th>ê</th>
<th>ó</th>
<th>ç</th>
<th>à</th>
<th>Ê</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCRC code</td>
<td>e1</td>
<td>e2</td>
<td>e6</td>
<td>o2</td>
<td>c0</td>
<td>a3</td>
<td>E1</td>
</tr>
</tbody>
</table>

### 2.3 Machine readable phonetic symbol

Although IPA has complete definition of phonemes for a lot of languages in the world, computer can not easily read or display them without switch font settings. There exits machine readable phonetic symbol such as SAMPA [3, 4]. However, from our engineering experience, using “/ /”, “/ /”, “@”, “…” for machine reading/displaying will bring troubles in some kinds of platform and speech analysis tool during the multi-lingual
speech data processing. For this reason, we proposed a two-letter scheme of phonetic symbols during our multi-lingual TTS speech corpus. Because the idea of proposing this scheme is like to ARPAbets of English, we call it MCRCbets. Table 2.3 shows examples of some phoneme symbols with MCRCbets.

Table 2.3  Examples of MCRCbets for phonetic symbols

<table>
<thead>
<tr>
<th>IPA</th>
<th>∆</th>
<th>∩</th>
<th>%</th>
<th>≡</th>
<th>∆∪</th>
<th>∅</th>
<th>≡&lt;</th>
<th>:</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCRCbets</td>
<td>aa</td>
<td>oy</td>
<td>nj</td>
<td>sh</td>
<td>au</td>
<td>dd</td>
<td>ch</td>
<td>ae</td>
<td>ng</td>
</tr>
</tbody>
</table>

3  Lexicon creation

The roles of multi-lingual TTS lexicon are: (1) to calculate phonetic coverage for corpus design; and (2) to train or test letter-to-sound module.

It may be easy to find electronic lexicon with pronunciation for some major languages, including English, Mandarin, French and Spanish etc. However, for small-population languages such as Danish, it is hard to find such kind of resources. For some languages with regular pronunciation rules, such as Spanish, German, Italian, Portuguese, Russian and Arabic, pronunciation is always absent in lexicons. Some available electronic lexicons, for example LDC, is aimed for ASR, and usually don’t have stress position. For all of these reasons, TTS developer always needs to develop lexicon themselves.

3.1  Lexicon entries

To create a TTS lexicon, a big text corpus is necessary. There are great amount of free resources, such as web pages or HTML pages. The problem is to verify if the captured content is really in the language we want, because there are usually hyperlinks to HTML pages in other language.

The processing on downloaded materials includes text formatting, text normalization and counting occurrence. As a result the statistics on text can provide a word list and this list can be used as rough entries of a lexicon for target language. The main process is shown in Figure 3.1.

Manual lexicon checking needs native speaker’s help. Noisy items should be removed. Word pronunciation from rule-based prediction is not necessarily correct due to exceptional cases. An electronic lexicon can include more than one hundred thousand entries. It is inevitable to bring errors when manual work is used. So automatic checking is very important to interactive checking.
In practice, automatic letter-to-phoneme alignment is an effective and efficient approach to check error. About 2,000 words and their corresponding pronunciations can be aligned as seeds to align a complete lexicon. The conventional approach is to estimate the probability that one character $c$ corresponds to one phoneme $p$ using DTW (Dynamic Time Warping) algorithm [5]. This algorithm procedure can be modified to allow one character to correspond to two or more phonemes, or an epsilon (see Figure 3.2.1). A character $c_i$ can correspond to a phoneme sequence $p_t \cdots p_j$, if $t$ equals $j$, then $c_i$ corresponds to an epsilon. $i, j$ stand for character and phoneme respectively. A two-dimensional array $\text{Prob}[i][j]$ is to hold the maximum probability product from the starting point $(0,0)$ to the node $(i,j)$. Another array $\text{Path}[i][j]$ is to hold the optimal node sequence from the starting point $(0,0)$ to the node $(i,j)$. The induction process is presented as follows.

$$\text{Prob}[i][j] = \max\{\text{prob}[i-1][t] \times \text{prob}(p_t \cdots p_j \mid c_i), 1 \leq t \leq j\}$$  \hspace{0.5cm} (1)$$

where $\text{Pr ob}(p_t \cdots p_j \mid c_i)$ is the probability that $c_i$ corresponds to $p_t \cdots p_j$. The optimal sequence can be described as:

$$\text{Path}[i][j] = \text{Path}[i-1][k] \cup \{(i,j)\}$$  \hspace{0.5cm} (2)$$

where $k = \arg \max_i \{\text{Pr ob}[i-1][t] \times \text{Pr ob}(p_t \cdots p_j \mid c_i), 1 \leq t \leq j\}$

Several iterations can arrive at an accurate aligned lexicon, or letter/letter alignment information. We can check items in lexicon very quickly by looking for
exceptional cases. For example, we once employed alignment information to correct more than 6000 entries for a UK English lexicon.

![Alignment samples of fax and cakes](image)

**Figure 3.2.1** Alignment samples of *fax* and *cakes*

## 4 Script design and phonetic coverage

Typical variable-length speech unit selection based concatenative TTS needs a speech corpus with abundant phonetic phenomena and there is no limitation on size. For this purpose, speech unit coverage for corpus design can be frequently used phrase, word, syllable string, syllable, phone string [6, 7] and etc.. Speech unit inventory for coverage results from statistics based on big text corpus and pronunciation lexicon.

Prosodic structure in an utterance may have some relationship with syntactic, semantic and physical constraints. Unfortunately, prosody structure cannot be predicted accurately at text level. With the same text, different speaker may use different strategies of phrasing or breaking. A simple way is to let target speech units appear at different positions in an utterance, such as initial, middle and final. For sentence middle, more times occurrence is helpful for coverage. It is perfect when all target speech units occur in different prosody position. Greedy algorithm [8] is always used for sentence selection (see figure 4).

To synthesize colorful speech, expressive sentence and more linguistic phenomena should be comprised. The major cases are: (1) Statement sentence: Most text is statement style. These kinds of text should be the principal part. Most of phonetic occurrence is in this part; (2) Question sentence: There are two kinds of question sentences: with or without question word. The major question words are: when, where, which, what, who, whom, how, etc… (3) Expressive sentence: The expressive sentences are always characterized with special punctuation (! …), and modal words. (4) numbering: There are numbering plan when speaker read digital string. For example, telephone number will be grouped based on national area, local area and so on; (5) appellation; (6) Dialect features, such as retro-flex in Mandarin.
5 Speaker selection

Speaker selection is an essential and critical step in the development of all concatenation based TTS systems. The quality of original utterances that are recorded in TTS speech corpus by native and professional speaker directly determines the final performance or success of TTS engine. Quality may include recording condition, timbre and speaking style of speaker etc. Effective and quantitative quality control methods corresponding to each aspect are required and helpful for speaker selection efficiently.

Neutral-like speaking style is preference for concatenation based embedded system. It means that the fluctuation of acoustic features (such as pitch, duration and magnitude) of utterance or speech unit should be limited into a small range. Here pitch distribution is employed to measure the pitch fluctuation of each candidate. In
the following figure, the pitch distribution of three candidates is obtained with statistical analyzing. Among three pitch curves, speaker-2 will be an ideal candidate due to his minimum pitch fluctuation.

![Pitch Fluctuation Measure](image)

Figure 5 Three speakers’ pitch distributions

6 Speech recording

The recording is carried on in a sound proof room (Reverberation < 0.3s). MOTO-Recorder is recording software with 2 synchronized channels where channel one for a large membrane microphone (AKG K4000B, 20Hz-20KHz) and channel 2 for the laryngograph. All signals are sampled synchronously through a mixture (Spirit Live 4) and a soundcard (ESI WaMi Rack 192X) and stored directly to hard disc. For each sentence, the speakers can utter it repeatedly until they are satisfied with their pronunciation; however all the mispronounced or unsatisfied utterances are backed up at the same time. Be careful to mount the laryngograph and find the optimal mounting position to get high quality signals. Figure 6 shows the microphone and Laryngograph mounting positions and the recording setup map.

Many factors affecting the recorded speech quality have to be controlled such as speech rate, SNR, the stability of the voice and even the speaker’s emotion. For bilingual speakers who have to record two different languages for building up a bilingual TTS system, we try to control both languages’ speech rates and pitch registers on a same level. This can be done by recording two languages alternately rather than recording one by one. After recording, the speakers are asked to check all their own utterances carefully and try to select those defective ones. Then the speakers have to re-record these sentences.
7 Speech segmentation

In general, the HMM-based frameworks exhibit superior performance compared with other approaches for automatic speech segmentation [9]. With an accompanying orthographic transcription, tri-phone HMMs trained by HTK toolkit [10] were adopted in this paper. The number of states was two for stops, pause and silence, and three for other phonemes, and the number of mixtures for each state was set to 16. The parameter vectors comprised 12th order MFCC parameters and energy, plus delta and acceleration coefficients (total 39 coefficients). The frame size was 20 ms with 5 ms frame shift. Although the results acquired from standard HMM-based approach are quite impressive, there are also many shortcomings [11] that prevent it from achieving prefect performance. Therefore, improved process should be incorporated. The block diagrams of the segmentation process are illustrated on Figure 7. The input is the speech signal and its word level transcript and the output is the phonetic sequence.

The text-to-phoneme conversion was implemented using the corresponding phonetic dictionary of word pronunciations. In the word-based alignment, the phonetic spelling of each word is determined using a dictionary lookup. As for the words which do not appear in the dictionary, letter-to-sound rules can be used to estimate the word pronunciations. After acquiring the first trained HMMs, labels correction process using automatic recognition is need to account for phonological changes, which naturally occur in continuous speech, such as English phrase “what time” would be transcribed as /w aɪ t aɪ m/. Moreover real pause and silence labels may be inserted by comparing the determined duration with a predefined threshold and multiple pronunciations are checked. The last step of the segmentation process is the phoneme boundary correction, which uses zero-cross rate, energy, pitch and physical information.
To evaluate the effects of the whole process, the output before the boundary correction step is extracted for the manual correction and then the results are set as the reference. The segmentation accuracy are calculated as the tolerance in the time shift between the determined boundaries and the reference boundaries within 10 and 20 ms. Without the boundary correction process, the average boundary error is 14.2 ms and the total percentage in agreement within 10 and 20 ms are 76.3% and 90.2% respectively. By applying boundary correction process, such as speech signal processing using glottal signal and other information, the average boundary error decreased to 8.3 ms, and the total percentage in agreement increase to 86.4% within 10 ms and 95.5% within 20 ms.

8 Prosody annotation

Before annotating a corpus for a specific language, the annotation agreement among 3 trained professional transcribers is always rated until a high agreement rate more than 80% is reached. The annotation tool is Praat (http://www.praat.org/). For English corpus, pitch accent and phrase accent, intermediate phrase and intonation phrase boundaries are annotated as show in Figure 8. For Chinese corpus, prosodic word and prosodic phrase boundaries, and prosodic word and phrase stress are annotated. For other languages, two level prosodic boundaries are annotated.

Figure 7 Block diagrams of the segmentation process
Figure 8 An annotated utterance “For sheer marketing power, our baker had no peer”, in which BI stands for the break index tier, 3 for intermediate phrase boundary (pitch accent ID as in IViE [12]), 4 for intonational phrase boundary; SI for pitch Accent stress tier annotated on the prominent vowels, and BT for the boundary tone tier.

9 Discussion

Speech corpus design, data collection and data processing are the fundamental issues of TTS system development. Any misplay in procedure of building speech corpus will cause TTS quality problems. In practice, although data processing management scientifically can reduce language complexity, native speakers’ involvement is necessary particularly in pronunciation checking. It is also necessary for TTS developers to acquire some language knowledge. For native speaker may not understand the differences between linguistics and speech engineering, checking their work become important. The interactive check by automatic methods is efficient in multi-lingual TTS speech corpus development.

References