G2P-FREE GRAPHEME-TO-SPEECH SYNTHESIS: UTF-8 BASED AUTOMATIC UNIT-DATABASE ANNOTATION AND MIXED VITERBI UNIT-SELECTION

Mythri Thippareddy[*], *C. Mahima, S. Adithya, Sunil Rao, V. Ramasubramanian*[**]

PES Institute of Technology - Bangalore South Campus (PESIT-BSC)
Bangalore, India
[*]mythri.thippareddy@gmail.com, [**]v.ramasubramanian@pes.edu

ABSTRACT

We propose a grapheme-based speech synthesis solution without requiring a G2P for the language Kannada, one of the 22 scheduled languages in India, by exploiting the fact that it has a highly phonemic orthography. We show that its underlying UTF-8 representation can be mapped to unique monophones with minimal grapheme processing. Under such a UTF-8 representation of the unit-database orthography and the input text to be synthesized, we propose various formulations towards realizing a fully UTF-8 based speech synthesis: i) a template based segmental K-means (SKM) algorithm for annotation of the unit-database directly in terms of UTF-8 code, ii) UTF-8 based unit-selection in several variants, namely, a unified Viterbi framework generalizing to both monophone and syllabic representation of the unit-database and the input text, a grouped Viterbi for monophone representation allowing unit-selection with very low complexity and, a joint monophone-syllabic Viterbi with optimal backoff to monophone units. We use a double-ended objective quality measure of the synthesized speech in terms of spectral distortion and compare the proposed UTF-8 based system with several state-of-art TTS approaches and show that the proposed UTF-8 based grapheme-to-speech synthesis of Kannada is a highly viable solution.

Index Terms— GTP-free TTS, grapheme-to-speech, UTF-8 segmental K-means, UTF8 segmentation and labeling, UTF-8 Viterbi, optimal backoff unit-selection

1. INTRODUCTION

Text-to-speech synthesis, in its conventional form, depends crucially on grapheme-to-phoneme (G2P) conversion, which in turn has varying degrees of complexity depending on whether the language has shallow or deep orthography, with the depth of the orthography representing the degree to which it diverges from being truly phonemic. Highly phonemic orthography allows one-to-one correspondence between the graphemes (written symbols) and the phonemes (spoken sounds) of the language. A classic example of a language with deep orthography (or which is highly non-phonemic) is English, calling for a complex G2P apart from usage of extensive pronunciation lexicon or dictionaries.

On the other hand, there do exist several languages (a good deal of them among the Indian languages) which have highly phonemic orthography – Kannada being a classic example. Kannada is a Dravidian language spoken predominantly by approximately 40 million native Kannada people in the South Indian state of Karnataka, and ranks 33rd in the list of most spoken languages in the world. The fact that Kannada has a highly phonemic orthography (i.e., its script is almost perfectly phonetic) allows the possibility that a TTS system for Kannada can be entirely made from its graphemic representation (i.e. its underlying UTF-8 codes) rather than require a full-fledged G2P module. The prime motivation towards exploring such a possibility is to be able to i) circumvent the G2P and associated phone-level representation of the unit-database text and speech, complex segmentation and labeling schemes, such as is examined in some recent state-of-art unit-selection systems [1, 2], and ii) being able to derive a flexible and unified Viterbi based unit-selection formulation capable of work on monophone and syllabic UTF-8 units individually or jointly, the latter leading to a pleasing optimal back-off strategy within an elegant Mixed-Viterbi unit-selection framework.

In this paper, we examine these issues and propose various algorithms towards a fully UTF-8 end-to-end grapheme-based speech synthesis for Kannada: i) minimal grapheme processing algorithms to realize a UTF-8 representation of the unit-database in terms of both monophone and syllabic units, ii) a fully automatic segmental K-means (SKM) algorithm for segmentation and labeling of the unit-database in terms of UTF-8 based units (both monophone and syllabic), iii) a unified Viterbi unit-selection algorithm generalizing to both monophone and syllabic representations of the unit-database and, iv) a joint monophone-syllabic Viterbi unit-selection which allows an optimal back-off to monophone units when the syllabic coverage of the unit-database is inadequate for arbitrary input text. We use the double-ended objective measure in terms of 'spectral distortion', recently proposed by us [6] for characterizing the synthesized speech quality and compare the proposed solutions with three other segmentation solutions (all further used in the Festival unit-selection synthesis platform) namely, the EHMM (the in-built embedded re-estimation based HMM segmentation and labeling tool of Festival), SKM algorithm for segmentation and labeling in terms of acoustic monophone labels derived by a conventional G2P converter, the Hybrid segmentation and labeling algorithm (which combines semi-automatic group-delay based initial segmentation with an iterative syllable-conditioned embedded re-estimation of monophone units [1,2]), and show that the proposed UTF-8 segmentation, labeling and unit-selection solution offers comparable performance, while being associated with a pleasingly low complexity and direct solution for phonetic language such as Kannada, without the inherent high complexity of (and/or diverse processing steps in) the other three solutions stated above and compared here.

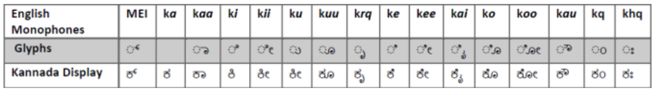
2. KANNADA LANGUAGE SPECIFICS

We outline here briefly, some of the specific characteristics of Kannada that allow the possibility of working with the underlying

UTF-8 codes for a fully grapheme-to-speech synthesis without a full-fledged G2P module.

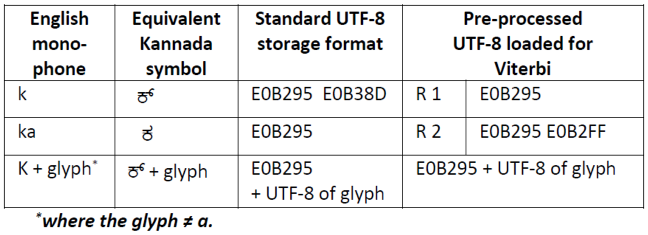
Kannada's writing system is essentially syllabic, in the sense that it has a syllabic alphabet, or alphasyllabaries, consisting of symbols for consonants and vowels. The consonants are combined with a vowel that is changed by special (e.g. diacritic or glyph) signs. Vowels are written separately, only if they occur at the beginning of a word or by themselves. In conjunction with a consonant in the middle of words, the consonants are modified by glyphs.

Kannada has 49 phonemic letters, 13 of which are vowel letters, 34 consonant letters, and 2 which are neither consonant nor vowel. The following is a generic table for a consonant /k/ and how it gets modified by the various possible 'glyphs' to represent the adjoining vowels.



English Monophones	MEI	ka	kaa	ki	kii	ku	kuu	krq	ke	kee	kai	ko	koo	kau	kq	khq
Glyphs	◌್		◌ಾ	◌ಿ	◌ೀ	◌ು	◌ೂ	◌ೃ	◌ೆ	◌ೇ	◌ೈ	◌ೊ	◌ೋ	◌ೌ	◌ಂ	◌ಃ
Kannada Display	ಕ್	ಕ	ಕಾ	ಕಿ	ಕೀ	ಕೀ	ಕೂ	ಕೃ	ಕೆ	ಕೇ	ಕೈ	ಕೊ	ಕೋ	ಕೌ	ಕಂ	ಕಃ

The following table shows an example of converting the symbolic script into its corresponding UTF-8 codes, and how they get associated with the underlying acoustic monophone units via a minimal set of two grapheme-processing rules. The two rules, marked R1 and R2, can be essentially viewed as what can be called 'virama-deletion' and 'implicit-vowel-(schwa)-addition' steps, respectively, aimed towards establishing a consistent mapping between the underlying UTF-8 and the associated acoustic realization of the text. Note that the 3rd row does not need any rule for conversion (for non-schwa vowels), and the input UTF-8s are retained as is. A minimal set of similar rules further group UTF-8s into syllabic units needed in Sec. 4 and 5.



English mono-phone	Equivalent Kannada symbol	Standard UTF-8 storage format	Pre-processed UTF-8 loaded for Viterbi	
k	ಕ್	E0B295 E0B38D	R 1	E0B295
ka	ಕ	E0B295	R 2	E0B295 E0B2FF
K + glyph*	ಕ್ + glyph	E0B295 + UTF-8 of glyph	E0B295 + UTF-8 of glyph	

***where the glyph ≠ a.**

3. G2P-FREE GTS SYSTEM

A conventional TTS system is typified by the following components: a phone-set, G2P, segmentation and labeling of a unit-database (annotation) in terms of the phone-set, unit-selection of input phone-sequence using the database, concatenative synthesis of selected units, with or without prosodic modifications. In contrast, a grapheme-to-speech (GTS) is different from a conventional TTS system in several respects: i) the unit-database is segmented and labeled in terms of UTF-8 code, with the important assumption that a distinct UTF-8 code consistently represents the same acoustic unit in all its many realizations across the unit-database, ii) the input text is also represented in terms of the UTF-8 code, and it is possible to synthesize speech by selecting the acoustic-units in the unit-database by means of matching the UTF-8 codes of the input text to the UTF-8 codes of the unit-database, by an appropriate unit-selection algorithm and, iii) the unit-selection algorithm performs this step directly on the UTF-8 code sequence (of the input text) against the UTF-8 code sequence (of the unit-database).

4. UTF-8 BASED SKM FOR AUTOMATIC SEGMENTATION

As noted above, one of the stringent requirement in a GTS system is that the unit-database can be segmented and labeled in terms of UTF-8 code, with the important assumption that a distinct UTF-8 code consistently represents the same acoustic unit in all its many realizations across the unit-database. In this work, we propose a segmental K-means algorithm which operates on the UTF-8 representation (label sequence) in a forced alignment mode to derive the segmentation and labeling of the unit-database in terms of the UTF-8 labels, both in terms of monophone and syllabic units.

The segmental K-means (SKM) algorithm is a template-based equivalent of the conventional HMM based embedded re-estimation typically used for HMM model training from continuous speech (given the orthography) or automatic segmentation (again, given the orthography of the speech to be segmented and labeled), the latter termed EHMM as used in Festival. The SKM algorithm proposed here is an iterative realization of two steps:

a) a forced alignment by multi-template one-pass DP segmentation and,

b) a modified K-means (MKM) algorithm to derive a small set of template codebook per phone class from the forced alignment segmentation clusters.

Like the EHMM, the SKM starts with a flat start of UTF-8 units derived from the unit-database orthography, and represents a 'seed-less' procedure for template based segmentation. By this procedure, we realize an accurate modeling of the underlying acoustic phonetic units, and further obtain accurate segmentation due to the iterative nature of refining the templates modeling the units and the successive forced alignments. We have obtained convergence characteristics of the SKM algorithm (the forced alignment error and MKM cluster variance, on 1000 Kannada sentences, and these show a monotonic and consistent reduction (by a factor of 2, starting from the flat-start values) until convergence in 10 iterations.

5. UTF-8 UNIT-SELECTION – VITERBI VARIANTS

5.1. Unified Viterbi unit-selection

Consider a 'continuous codebook' $\mathcal{U} = (u_1, u_2, \dots, u_n, \dots, u_N)$ which is essentially a sequence of UTF-8 vectors as occurring in continuous speech, being composed of N variable length UTF-8 units $(u_1, u_2, \dots, u_n, \dots, u_N)$, where a unit u_n is of length l_n monophone UTF-8 units or vectors, given by $u_n = (u_n(1), u_n(2), \dots, u_n(l_n))$, and with each UTF-8 vector defined in terms of three components, namely, the labels of primary UTF-8 (representing its phonetic identity), left-context UTF-8 and right-context UTF-8, (and possibly other features as could be extracted or predicted from the input text).

The unit-database is said to be made of monophone units, if $l_n = 1, \forall\, n = 1, \dots, N$, i.e., each unit is a monophone unit. The codebook is said to be made of 'syllabic' units if l_n is variable over n. When the unit-database is made of monophone units, we refer to the corresponding Viterbi unit-selection as 'Monophone-Viterbi' and when the unit-database is made of syllabic-units, we refer to the corresponding Viterbi unit-selection as 'syllabic-Viterbi'. The unified formulation (developed for speech coding by us earlier [3]) given below generalizes to both such unit-databases. This is in contrast to the conventional unit-selection framework, where the unit-database is treated as being made of monophone like units.

Let the input text which is to be synthesized, using the above unit-database, be a sequence of UTF-8 vectors $O = (o_1, o_2, \dots, o_t, \dots, o_T)$. Unit-selection, for purpose of concatenative synthesis, in its most general form involves segmenting and labeling this sequence of vectors O by a 'decoding' algorithm which optimally segments the sequence and represents (and synthesizes) each segment by an appropriate unit from the unit database.

Consider an arbitrary segmentation of O into a sequence of K segments $S = (s_1, s_2, \dots, s_{k-1}, s_k, \dots, s_K)$ with corresponding segment lengths $(L_1, L_2, \dots, L_{k-1}, L_k, \dots L_K)$. This segmentation can be specified by the segment boundaries $B = \big((b_0 = 0), b_1, b_2, \dots, b_{k-1}, b_k, \dots, (b_K = T)\big)$, such that the k^{th} segment s_k is given by $s_k = \big(o_{b_{k-1}+1}, \dots, o_{b_k}\big)$. Let each segment be associated with a label from the unit database, with each index having a value from 1 to N; let this index sequence be $Q = (q_1, q_2, \dots, q_{k-1}, q_k, \dots, q_K)$. The optimal unit-selection algorithm solves for (K^*, B^*, Q^*) so as to minimize an overall decoding distortion given by

$$D^* = \min_{K,B,Q} \left[(1-\alpha) \sum_{k=1}^{K} D_u\big(s_k, u_{q_k}\big) + \alpha \sum_{k=2}^{K} D_c(q_{k-1}, q_k) \right] \tag{1}$$

Here, $D_u\big(s_k, u_{q_k}\big)$ is the unit-cost in quantizing segment s_k using unit u_{q_k}. $D_c(q_{k-1}, q_k)$ is the concatenation-cost when unit $u_{q_{k-1}}$ is followed by unit u_{q_k}, which is given by

$$D_c(q_{k-1}, q_k) = \beta_{k-1,k} \cdot d\big(u_{q_{k-1}}, u_{q_k}\big)$$

where, $d\big(u_{q_{k-1}}, u_{q_k}\big)$ is the weighted Hamming distance between the last UTF-8 vector of unit $u_{q_{k-1}}$ and the first UTF-8 vector of unit u_{q_k}. $\beta_{k-1,k} = 0$, if $q_k = q_{k-1} + 1$ (i.e., $u_{q_{k-1}}$and u_{q_k} are consecutive in the database) and $\beta_{k-1,k} = 1$ otherwise. This favors selecting two consecutive units from the unit database to correspond to two consecutive segments in the input UTF-8 sequence.

We give here the Viterbi unit-selection algorithm to solve the above optimal decoding problem of Eqn. (1) for the general case that the units are monophone units or syllabic units. The Viterbi algorithm, (with the within-unit and cross-unit recursions, as given below) are computed time-synchronously on a trellis defined by the unit database $\mathcal{U}$ of N units on the y-axis (from which the unit-selection is done) and the given input text of T UTF-8s $O = (o_1, o_2, \dots, o_t, \dots, o_T)$ in the x-axis.

Given below are the dynamic program recursions of the unified Viterbi based unit-selection. The recursions are in two parts: within-unit recursion and cross-unit recursions.

Within-unit recursion

$$D(t, l, n) = D(t-1, l-1, n) + (1-\alpha) \cdot d_u(o_t, u_n(l))$$

Cross-unit recursion

$$D(t, 1, n) = \min_{r=1,\dots,N} [D(t-1, l_r, r) + \alpha \cdot d_c(r, n)] + (1-\alpha) \cdot d_u(o_t, u_n(1))$$

Here, the above two recursions are applied over all UTF-8s of all the units in the unit-database for every UTF-8 o_t of the input utterance $O = (o_1, o_2, \dots, o_t, \dots, o_T)$. The within-unit recursion is applicable only in the 'Syllabic Viterbi' unit-selection, where the syllabic-unit database has units made of a sequence of UTF-8 vectors, in the form of $u_n = \big(u_n(1), u_n(2), \dots, u_n(l), \dots, u_n(l_n)\big)$, i.e., $l_n \geq 1, \forall\, n$. Thus, the within-unit recursion is applied to all UTF-8 vectors in a unit which are not the starting UTF-8s, i.e., $l \neq 1$. The cross-unit recursion is applied only for the starting UTF-8s of all units, i.e., for $l = 1$ in the case of 'Syllable-Viterbi' to account for a potential entry into unit u_n from the last UTF-8 $u_r(l_r)$ of any of the other units $\{u_r\}, r = 1, \dots, N$ in the unit database. For the 'Monophone Viterbi', which operates on a monophone-database, i.e., with $l_n = 1, \forall\, n$, the within-word recursion is not applied, and only the cross-unit recursion exists to account for transitions from one (single UTF-8) unit to another.

$D(t, l, n)$ is the minimum accumulated distortion by any path reaching the grid point defined by UTF-8 o_t of the input text and UTF-8 $u_n(l)$ of unit u_n in the unit database. $d_u(o_t, u_n(l)$ is the unit cost between UTF-8 vector o_t of the input text and UTF-8 $u_n(l)$ of unit u_n. In the cross-unit recursion, (applicable to both Monophone-Viterbi and Syllabic-Viterbi), the term $d_c(r, n)$ stands for the concatenation-cost between the last UTF-8 $u_r(l_r)$ of unit u_r and the first UTF-8 $u_n(1)$ of unit u_n in the case of Syllabic-Viterbi, which reduces (as a special case) to the concatenation-cost between the single-UTF8 unit u_r and the single-UTF8 unit u_n in the case of Monophone-Viterbi. $(1-\alpha)$ and α respectively weigh the unit-cost and concatenation cost, thereby realizing Eqn. (1) and providing a parameter for controlling the relative importance of the two costs in determining the optimal path. The final optimal distortion (as in Eqn. (1)) is obtained as,

$$D^* = \min_{n=1,\dots,N} D(T, l_n, n)$$

The optimal number of segments K^*, segment boundaries B^* and segment labels Q^* (corresponding to this optimal D^* in Eqn. (1)) are retrieved by back-tracking.

5.2. Grouped Monophone-Viterbi unit-selection

The Monophone-Viterbi has a high complexity of $O(N^2T)$, for a unit-database size N and number of input UTF-8 indices T. This complexity can be reduced significantly for Monophone-Viterbi, by observing that each of the input units (here UTF-8 o_t) needs to be matched with only those units that have the same primary label in their UTF-8 vector as in the UTF-8 vector of o_t, i.e., o_t is matched with units in a group G_t which has all units in the unit database $\mathcal{U}$ that have the same primary (phonetic) label as o_t. The resulting complexity is now $O(|G_t|^2T)$, with size of a group $|G_t| \ll N$. Note that this does not compromise on the resultant solution, as it preserves the unit-cost to be the same as from the entire unit-database $\mathcal{U}$, but permitting high reduction in computational complexity of unit-selection.

We used this approach recently [5], and adopt it for the Monophone-Viterbi algorithm here. Note that the same approach cannot be used for the Syllabic-Viterbi, since the syllables in the input UTF-8 sequence are not defined a priori (unlike the availability of such information for Monophone-Viterbi decoding), and any attempt to define the syllable units a priori on the input UTF-8 sequence would render it sub-optimal, by way of having imposed a strict syllabification prior to Syllabic-Viterbi. Henceforth, ‘Monophone-Viterbi’ refers to this grouped-Viterbi approach, and Syllabic-Viterbi uses no grouping and is as described in Sec. 5.1.

5.3. Joint Monophone-Syllabic (or Mixed) Viterbi with optimal backoff

In Sec. 5.1, we presented a generalized Viterbi unit-selection which handles monophone units or syllabic units in a unified manner, respectively termed the ‘monophone-Viterbi’ and ‘syllabic-Viterbi’. Note that each of these exclusively work as a monophone-only or a syllable-only decoding of the input UTF-8 sequence, and selects units that are only monophones (and corresponding concatenation and synthesis thereof) or that are only syllables (and corresponding concatenation and synthesis thereof).

The monophone-only decoding is optimal in the sense of never incurring ‘substitution errors’, i.e., it ensures that each monophone unit in the input UTF-8 sequence is exactly matched to the units selected with respect to the primary phonetic identity, though possibly not ensuring a left-context / right-context match also. By this, the synthesis retains high fidelity in terms of the phonetic constitution of the input text.

On the other hand, the ‘syllable-only’ decoding of the ‘Syllabic-Viterbi’ selects syllabic-units (made of several monophone units) to synthesize an input UTF-8 sequence, which could also be contiguous, resulting in long sequences of naturally contiguous acoustic unis to be synthesized, thereby preserving co-articulation, both within the syllabic-units and across contiguous syllabic-units.

However, the Syllabic-Viterbi suffers from its potential to cause substitution errors due to inadequate ‘syllable-coverage’ in the unit database, i.e., for a limited number of syllables in the unit-database (which is usually likely to be the case, considering the combinatorially large number of syllables required to exhaustively cover all possible syllables that can occur in the input text, or in the large number of possible ‘syllabification’ of the input text). This primarily arises from the way the decoding is done – the input UTF-8 sequence (on the x-axis) is segmented and labeled in terms of the syllabic units in the unit-database (on the y-axis), and hence the optimal Viterbi path can result in a particular segment of the input UTF-8 sequence (in the x-axis) to be forcibly mapped to some syllable in the unit-database (in the y-axis) in such a way that the selected syllabic unit may have a monophone (acoustic unit) level substitution error in the sense that a monophone in the input segment (in the x-axis) does not have the same monophone in the selected unit (on the y-axis)..

In order to retain the advantages of the syllable-only Syllabic-Viterbi decoding (in terms of realizing syllabic decoding, ensuring preserving within- and across-syllable co-articulations), but without the associated substitution error problems (arising from the inevitable inadequate syllable-coverage problem), we now propose a ‘Mixed-Viterbi’ decoding solution.

Here, the monophone units and syllabic units are combined into a single database, i.e., the same unit-database is indexed in terms of monophone units as well as syllable units and made available on the y-axis. The ‘Mixed Viterbi’, operating on such a database, performs an optimal ‘backoff’ to mono-phone units, whenever the input UTF-8 sequence (derived from the input text to be synthesized) cannot be adequately decoded in terms of only the syllables (as with a ‘syllable-only’ decoding). By this, the substitution errors are minimized, and, as we will show, for correct trade-offs between unit-cost and concatenation-cost, the substitution error can even be completely eliminated, thereby offering the best of both solutions – the syllabic unit-selection with inherent preservation of within-syllable co-articulation, and not incurring any substitution errors, as was ensured in a monophone-only unit-selection (but which was shown to offer poorer preservation of co-articulation or unit to unit contiguity).

We outline the basic formulation of the ‘mixed Viterbi’ unit-selection now. Let the ‘composite’ unit-database be represented as $\mathcal{D} = \mathcal{S} + \mathcal{U}$, as a combination of two databases: a syllabic-unit database $\mathcal{S} = (S_1, S_2, \ldots, S_j, \ldots, S_N)$ of N syllabic units, each denoted by upper-case S (and not to be confused with the lower-case s used to denote the segments of input UTF-8 sequence in Sec. 5.1), and the “same database” indexed as a monophone-unit database $\mathcal{U} = (u_1, u_2, \ldots, u_n, \ldots, u_M)$ of M single-UTF8 (monophone) units (with $M > N$). For a given input text, with UTF-8 sequence $O = (o_1, o_2, \ldots, o_t, \ldots, o_T)$, the desired kind of decoding using the composite database, if a part of the input sequence e.g., $(\ldots, o_{11}, o_{12}, o_{13}, \ldots)$ cannot be mapped to a syllabic unit in the syllabic-unit database $\mathcal{S}$, (due to non-availability of such a syllable in this database), then the optimal Mixed-Viterbi path will take recourse to mapping these units using some monophone units, e.g. $(u_{103}, u_{25}, u_{97})$ from the monophone-unit-database $\mathcal{U}$. Note that in the absence of the availability of the monophone-unit database, the input would have been forcibly decoded (segmented and labeled) by the Syllable-Viterbi (i.e., in terms of ‘only’ syllable units), by the closest syllabic units, thereby incurring a ‘substitution’ error of these three units, when they are assigned some syllabic-unit(s) which are not made up of the same constituent monophone units. It can thus be seen, that in general, such a ‘provisioning’ of the monophone-unit database $\mathcal{U}$ along with the syllabic database $\mathcal{S}$ helps in retaining the co-articulatory advantages of syllabic units, even while avoiding incurring substitution errors (which reduce the fidelity of synthesized speech) arising from inadequate syllable coverage (for any given practical sizes of the syllabic database $\mathcal{S}$), by mapping monophones UTF-8s in the input sequence to monophone units in the monophone-unit database $\mathcal{U}$, and recovering an optimal Viterbi path that ensures the highest fidelity (or lowest unit-to-unit unit-costs along the path), within the desired concatenation constraint determined by the controlling parameter α.

We now give the details of such a mixed-Viterbi unit-selection with optimal back-off, given a composite database $\mathcal{D}$. In the formulation given below, we use the Grouped Monophone-Viterbi formulation (Sec. 5.2) which uses a grouped monophone unit database $\mathcal{G}$, which is made of T groups $(G_1, G_2, \ldots, G_{t-1}, G_t, \ldots, G_T)$, as indicated in Sec. 5.2, where a group G_t is made of units from $\mathcal{U}$ that have the same primary UTF-8 label (phonetic identity) as the primary UTF-8 of the input UTF-8 vector o_t.

Let the Mixed-Viterbi consider a path that maps the input UTF-8 sequence $O = (o_1, o_2, \ldots, o_t, \ldots, o_T)$ to a sequence $Q = (q_1, q_2, \ldots, q_{t-1}, q_t, \ldots, q_T)$, so that each UTF-8 o_t maps to some unit u_{q_t} in the composite database $\mathcal{D}$, i.e., while such a generic path is actually a sequence of syllabic units drawn from $\mathcal{S}$, typically viewed as interspersed with monophone units drawn from $\mathcal{U}$, it can further be represented in terms of the ‘all’ monophone UTF-8 units by expanding each syllable in the path in terms of its constituent monophone units; for instance, the syllabic-unit S_j is viewed as made of l_j “contiguous” monophone units given by, $S_j = \left(u_j(1), u_j(2), \ldots, u_j(l), \ldots, u_j(l_j)\right)$, where $u_j(l)$ is some monophone unit in the same unit-database, indexed by some other sub-script in $\mathcal{U}$. The convenience of doing so is to facilitate a simple formulation of the overall distortion associated with the path Q, and in arriving at the mixed-Viterbi algorithm.

The overall distortion associated with such a path Q is given as,

$$D(O,Q) = (1-\alpha)\cdot\sum_{t=1}^{T} d_u\left(o_t, u_{q_t}\right) + \alpha\cdot\sum_{t=2}^{T} d_c\left(u_{q_{t-1}}, u_{q_t}\right) \tag{2}$$

Note that there are T unit-costs in the above distortion, and $(T-1)$ concatenation-costs corresponding to various types of unit to unit transitions. Specifically, the path Q can be made of 5 types of unit-to-unit transitions: within syllable (WS) unit transition, syllable-to-syllable (SS) unit transitions, monophone-to-syllable (MS) unit transitions, monophone-to-monophone (MM) unit transitions and syllable-to-monophone (SM) unit transitions. Accordingly, denoting the generic unit cost $d_c\left(u_{q_{t-1}}, u_{q_t}\right)$ simply as $d_c(t-1,t)$, we can expand it in the partitioned form of,

$$\begin{aligned}\sum_{t=2}^{T} d_c\left(u_{q_{t-1}}, u_{q_t}\right) &= \alpha_{WS}\sum_{\substack{u_{q_{t-1}}\in S_j\\ u_{q_t}\in S_j}} d_c(t-1,t)\\ &+\alpha_{SS}\sum_{\substack{u_{q_{t-1}}\in S_i\\ u_{q_t}\in S_j}} d_c(t-1,t) + \alpha_{MS}\sum_{\substack{u_{q_{t-1}}\in G_{t-1}\\ u_{q_t}\in S_j}} d_c(t-1,t)\\ &+\alpha_{MM}\sum_{\substack{u_{q_{t-1}}\in G_{t-1}\\ u_{q_t}\in G_t}} d_c(t-1,t) + \alpha_{SM}\sum_{\substack{u_{q_{t-1}}\in S_i\\ u_{q_t}\in G_t}} d_c(t-1,t)\end{aligned} \tag{3}$$

It is easy to note that $\alpha_{WS}=0$, since the within-syllable transition occurs within a syllable and should not be penalized, in order to preserve within-syllable co-articulation. The α in Eqn. (2) gets distributed as the other 4 αs in Eqn. (3). Considering the above expansion, the Mixed-Viterbi is made of the following recursions:

Within-Syllable Recursions

$$\begin{aligned}D(t,l,j) = D(t-1,l-1,j) + d_c^{WS}\left((u_j(l),(u_j(l-1)\right)\\ + d_u\left(o_t, u_j(l)\right)\end{aligned}$$

Into-Syllable Recursions

$$\begin{aligned}D(t,1,j) = \min\begin{bmatrix}\min\limits_{i=1,\ldots,N}\left[D(t-1,l_i,i) + d_c^{SS}(i,j)\right],\\ \min\limits_{u_k\in G_{t-1}}\left[D(t-1,u_k) + d_c^{MS}(k,j)\right]\end{bmatrix}\\ + d_u\left(o_t, u_j(1)\right)\end{aligned}$$

Into-Monophone Recursions

$$D\left(t,u_j\right) = \min\begin{bmatrix}\min\limits_{u_k\in G_{t-1}}\left[D(t-1,u_k) + d_c^{MM}(k,j)\right],\\ \min\limits_{i=1,\ldots,N}\left[D(t-1,i,l_i) + d_c^{SM}(i,n)\right]\end{bmatrix} + d_u\left(o_t,u_j\right)$$

In the above three recursions, the individual terms are as follows: In the within-syllable recursion, $D(t,l,j)$ is the accumulated minimum distortion of the best path reaching the trellis co-ordinate at index t of the input UTF-8 sequence and unit $u_j(l)$ in the syllable S_j. This can be reached only from unit $u_j(l-1)$ at time $(t-1)$, and incurs a Within-Syllable (WS) concatenation cost as given, which is actually 0. The Into-Syllable recursion applies to the first monophone UTF-8 of a syllable S_j, i.e., $u_j(1)$, with the corresponding minimum distortion denoted as $D(t,1,j)$, being reached from any one of the N other syllables (the first term in the outer min) or any one of the $|G_{t-1}|$ monophone units (the second term in the outer min), with their corresponding Syllable-to-Syllable (SS) or Monophone-to-Syllable (MS) concatenation costs. The Into-Monophone recursion applies to each of the M monophone units in $\mathcal{U}$, with the minimum distortion of the optimal path reaching monophone unit u_j at index t of the input UTF-8 sequence, with contention between every monophone unit in group G_{t-1} (first term inside the outer min) and each of the N syllables, i.e., from the last monophone unit $u_i(l_i)$ of syllable S_i (th second term inside the outer min), with their corresponding Monophone-to-Monophone (MM) and Syllable-to-Monophone (SM) concatenation costs. The unit cost in each of the above three recursions is self-explanatory.

6. EXPERIMENTS AND RESULTS

We now present results comparing the three Viterbi-variants proposed here: the Monophone-Viterbi, Syllabic-Viterbi and Mixed-Viterbi.

First, we focus on showing how the syllabic-Viterbi offers better mapping of the input UTF-8 sequence to syllabic units with high contiguity, with associated enhanced co-articulation, than Monophone-Viterbi which seems to offer only lesser contiguity and hence less natural co-articulation. Fig. 1 shows this in the form of 'contiguity-histograms' (at $\alpha=0.5$), which is the relative frequency distribution of the number of contiguous blocks of various sizes, as found in the decoded sequence of the three Viterbi variants. From this, it can first be noted that the syllabic-Viterbi has higher contiguity than Monophone-Viterbi, establishing that Syllabic-Viterbi is indeed the better choice, when compared to Monophone-Viterbi, in realizing better natural co-articulation in the synthesized speech. We show results obtained from a database made of 50 Kannada sentences spoken by a single Female speaker, and yielding unit-databases with M=4582 monophone units and N=2404 syllabic units together making up the composite unit-database.

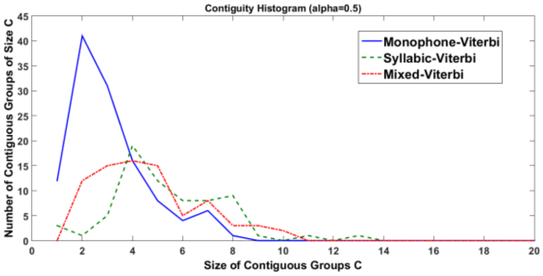


Fig. 1 Contiguity histograms for the three Viterbi variants.

Secondly, we show how the Mixed-Viterbi provides an automatic optimal back-off whenever the syllabic coverage is inadequate, and reduces i) substitution errors and ii) the overall spectral distortion, used here as an objective double-ended measure (proposed by us recently [4]). Fig. 2 shows the substitution errors for the three systems considered here. As alluded to in Sec. 5.3, the grouped Monophone-Viterbi does not incur any substitution errors; in contrast, as expected in the first paras of Sec. 5.3, the Syllabic-

Viterbi incurs substitution errors, which increase with increase in α, since the unit-cost gets weighed less $(1-\alpha)$ and results in favoring long concatenations at the expense of substitution errors. The Mixed-Viterbi reduces the substitution errors due to optimal 'back-off' to the monophone units, and thereby significantly reduces the substitution errors incurred by a forced syllabic decoding by the Syllabic-Viterbi system.

This can in turn be seen in the spectral-distortion (SD in dB) measure also in Fig. 3. The SD is obtained by using a reference of the synthesized speech, in the form of speech spoken by the same speaker as the unit-database from the given text being synthesized. This is obtained as an average over 10 Test text-phrases. The Monophone-Viterbi can be noted to have the least SD (by virtue of it not incurring any substitution error, and hence offering high fidelity to the phonetic content of the input text, and the reference speech). The Syllabic-Viterbi incurs a higher spectral distortion, at all values of α – at lower α, the unit-cost is weighed high, but nevertheless, a syllabic-only Viterbi invariably incurs substitution errors (as in Fig. 2) due to the monophone units within a syllable being mapped incorrectly to those in wrong syllabic units in the unit-database, due to inadequate syllable coverage. The SD performance of the Mixed-Viterbi can be seen to be substantially improved for these αs.

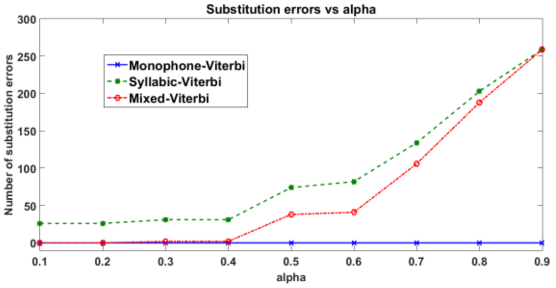


Fig. 2 Substitution errors for the three Viterbi variants vs α

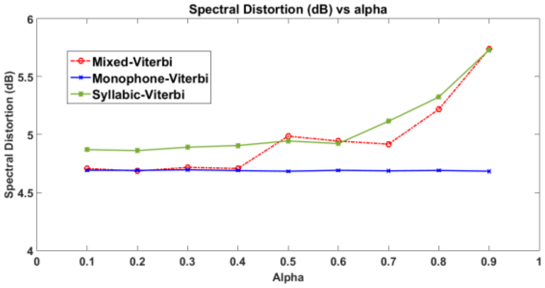


Fig. 3 Spectral-distortion (in dB) for the three Viterbi variants vs α

From these results, it also seems likely that an optimal weighting of $\alpha = 0.5$ is conducive for a balanced trade-off between unit-cost and concatenation cost, though a smaller α can also be preferred to realize the advantage of the Mixed-Viterbi over the other two Viterbi variants, with the advantage of preserving natural co-articulation (by means of syllabic-decoding), combined with no substitution errors (by means of back-off to monophone units, determined optimally via the Viterbi, without requiring any a prior heuristics).

Synthesis Results: We also compare the proposed three Viterbi-variants based on UTF-8 based unit-database segmentation/labeling and unit-selection with three other well established unit-selection systems (essentially, the Festival platform) using three different 'acoustic' based segmentation strategies for the unit-database annotation: EHMM (HMM based embedded re-estimation, part of Festival), segmental K-means approach on acoustic monophone units (SKM-A) and Hybrid segmentation (a group-delay and HMM based syllable conditioned embedded re-estimation approach [1,2]). Fig. 4 shows the spectral-distortion (SD) for these 6 systems across 10 Test text phrases used for synthesizing speech (the average of which was shown in Fig. 3). It can be noted that the three proposed Viterbi-variants compare well with the Festival based synthesis of the other three segmentation techniques; considering that the proposed Viterbi variants do not work on decision-tree optimized units or use sophisticated concatenative synthesis, they compare favorably to the more elaborate systems mentioned above, and indicated in this figure. Moreover, the Mixed-Viterbi can be seen to offer better performance than the Syllabic-Viterbi across most of the 10 sentences, contributing to the difference in the average SD shown in Fig. 3 for this value of $\alpha = 0.2$.

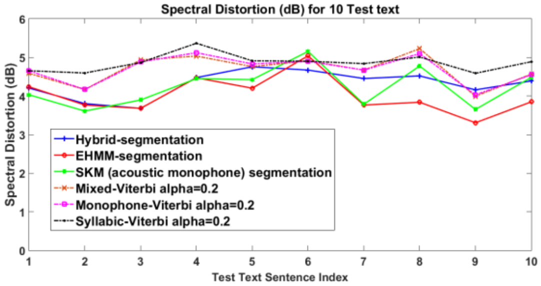


Fig. 4 Spectral-distortion (in dB) comparisons over 10 'Test' text

7. ACKNOWLEDGEMENTS

The authors thank the Department of Information Technology, Ministry of Communication and Technology, Government of India, for funding the TTS Consortium project, "Development of Text-to-Speech synthesis for Indian Languages Phase II", Ref. no. 11(7)/2011HCC(TDIL) under which this work was carried out.

8. REFERENCES

[1] S. Aswin Shanmugam, Hema Murthy. A hybrid approach to segmentation of speech using group delay processing and HMM based embedded reestimation. Proc. Interspeech '14, pp. 1648–1652, Singapore, Sep 2014.

[2] K. Raghava Krishnan et al., IIT Madras's Submission to the Blizzard Challenge 2014, Proc. Blizzard Challenge '14, Satellite workshop of Interspeech '14, Singapore, 2014 http://festvox.org/blizzard/blizzard2014.html

[3] V. Ramasubramanian and Harish Doddala. Ultra low bit-rate speech coding. SpringerBriefs in Speech Technology, Springer, 2015.

[4] Sunil Rao, C. Mahima, S. Vishnu, S. Adithya, A. Sricharan, V. Ramasubramanian, "TTS evaluation: Double-ended objective quality measures", Accepted in IEEE CONECCT '15, July 2015.

[5] T. Mythri and V. Ramasubramanian, "Prosody transplantation using unit-selection: Principles and early results", Accepted in IEEE CONECCT '15, July 2015.