Structural Data-Driven Prosody Model for TTS Synthesis

Jan Romportl

Department of Cybernetics
University of West Bohemia in Pilsen, Czech Republic
rompi@kky.zcu.cz

Abstract

This paper introduces a new data-driven prosody model for the text-to-speech system ARTIC. The model is intended to be almost language-independent and to generate naturally sounding intonation with a link to semantics. It is based on text parametrisation using a new prosodic grammar and on automatic speech corpora analysis methods. Its performance is evaluated by results of presented listening tests.

1. Introduction

Prosody is a very important element contributing to naturalness of synthetic speech and it is also an essential constituent of a spoken message structure. As a consequence, modelling of prosody has been already for a significant period of time treated as one of the crucial areas of text-to-speech system design.

The prosody model presented in the following text is conceptually similar to the approach of concatenative synthesis (enriched with a unit selection approach): it concatenates elementary prosody units derived from real speech data contained in a specially designed and annotated prosody corpus. The prosody units can have either one representant for a parametrisation of a specific portion of a text, or more representants and a prosody generation module chooses the best fitting one according to a particular criterion, as it is analogically in a TTS unit selection approach.

This data-driven prosody model can achieve significantly higher naturalness of resulting synthetic speech, similarly to the phenomenon when concatenative synthesis achieves higher naturalness than formant synthesis. This paper deals only with automatic fundamental frequency ($F0$) modulation implemented in the TTS system ARTIC [1], whereas questions of segmental duration and voice intensity are left open.

2. Text parametrisation

Each word (*prosodic word* respectively, see 2.1) of a sentence to be synthesised is assigned to a parametrisation vector D (we call it *description array*, DA) describing the word's functioning (in a linguistic sense) within the prosodic structure of the sentence. The parametrisation is based on the description of a sentence by a derivation tree (called *prosodic structure*) produced by the *prosodic grammar* when parsing the given sentence. This article presents the prosodic grammar very briefly, more information about it provides [2]. Justification for using such an abstract linguistic structural framework in machine prosody modelling can be supported by general conclusions of [3]. The idea of the formal prosodic structures and their constituents is based on the Czech classical phonetic view [4].

2.1. Prosodic grammar

The generative prosodic grammar consists of the following alphabet (symbols used in the grammar rules are parenthesised):

Prosodic sentence (PS)
Prosodic sentence is a prosodic manifestation of a sentence as a syntactically consistent unit, yet it can also be unfinished or grammatically incorrect.

Prosodic clause (PC)
Prosodic clause is such a linear unit of a prosodic sentence which is delimited by pauses. A prosodic sentence generally consists of more prosodic clauses.

Prosodic phrase (PP)
Prosodic phrase is such a segment of speech where a certain intonation scheme is realized continuously. A prosodic clause generally consists of more prosodic phrases.

Prosodeme (P0), (Px)
Prosodeme is an abstract unit established in a certain communication function within the language system. We have postulated that any single prosodic phrase consists of two prosodemes: so called "null prosodeme" and "functionally involved prosodeme" (where (Px) stands for a type of the prosodeme chosen from the list shown below), depending on the communication function the speaker intends the sentence to have. In the present research we distinguish the following prosodemes (for the Czech language; other languages may need some modifications):

P0 – null prosodeme; P1 – prosodeme terminating satisfactorily (P1-1 unmarked; P1-2 marked directive; P1-3 marked expressive; P1-4 specific); P2 – prosodeme terminating unsatisfactorily (P2-1 unmarked (supplementary); P2-2 marked declaratory; P2-3 marked disjunctive; P2-4 specific); P3 – prosodeme nonterminating (P3-1 unmarked; P3-2 marked bound; P3-3 specific)

Prosodic word (PW)
Prosodic word (sometimes also called phonemic word) is a group of words subordinated to one word accent (stress). Languages with a non-fixed stress position would need a stress position indicator too.

Semantic accent (SA)
By this term we call such a prosodic word attribute, which indicates the word is emphasised (using acoustic means) by a speaker.

There are two more terminal symbols used ("$" and "#") standing for pauses differing in their placement (inter- and intra-sentence). The terminal symbol (w_i) stands for a concrete prosodic word from a lexicon and $\emptyset$ means an empty terminal symbol. Note that Px is only an "abbreviation" for each prosodeme (i.e. P1-1, etc.). The rules should be understood this way: "$(PC) \longrightarrow (PP)\{1+\}\ \#\{1\}$" means that the symbol (PC) (prosodic clause) generates one or more (PP) symbols (prosodic phrases) followed by one $\#$ symbol (pause).

$$(PS) \longrightarrow (PC)\{1+\}\ \$\{1\} \quad (1)$$

$$(PC) \longrightarrow (PP)\{1+\}\ \#\{1\} \quad (2)$$

$$(PP) \longrightarrow (P0)\{1\}\ (Px)\{1\} \quad (3)$$

$$(P0) \longrightarrow \emptyset \quad (4)$$

$$(P0) \longrightarrow (PW)\{1+\} \quad (5)$$

$$(Px) \longrightarrow (PW)\{1\} \quad (6)$$

$$(Px) \longrightarrow (SA)\{1\}\ (PW)\{1+\} \quad (7)$$

$$(PW) \longrightarrow (w_i)\{1+\} \quad (8)$$

The grammar can be transformed into the Chomsky's normal form suitable for machine processing, yet the "intuitive" form shown above is more explanatory. Again, [2] explains what these rules mean in their relation to the language system.

2.2. Description function

Let P_S be a prosodic structure (i.e. derivation tree produced by the prosodic grammar) of a given sentence S. Let P_{NS} be a set of all nodes of P_S. For each node we can distinguish its type (e.g. (PS), (PP), etc.), type of its left and right neighbour (if there are any), the number of its neighbours, the number of its left neighbours (i.e. actually the index of the current node within its neighbours) and the link to its parent. From the theoretical point of view each node can be uniquely described (i.e. parametrised) by the *description function*,

$$\mathfrak{D} : P_{NS} \rightarrow \mathcal{D} \quad (9)$$

where $\mathcal{D} = P_{NS} \times P_{NS} \times N \times N \times \mathfrak{D}$ and N is the class of natural numbers. The recursion of this function is not a problem because all possible parsed trees are finite, although from the theoretical point of view we obtain an infinite-dimensional space. For each node $M \in P_{NS}$ we can determine

$$\mathfrak{D}(M) = (l_M, r_M, i_M, n_M, \mathfrak{D}(p_M)) \quad (10)$$

where l_M is the type of the left neighbour of M, r_M is the type of the right neighbour of M, i_M is the index of M within the scope of its neighbours, n_M is the number of neighbours of M and $\mathfrak{D}(p_M)$ is a description function of the parent node of M. If M is the root node, $\mathfrak{D}(p_M) = \emptyset$, which stops the recursion.

There is an ad hoc definition of the description function specially for the terminal symbols (i.e. leaf nodes). This function additionally includes intonationally relevant structural features of a prosodic word:

$$\mathfrak{D}_T(M) = (n_{pM}, n_{sM}, i_{sM}, l_M, r_M, i_M, n_M, \mathfrak{D}(p_M)) \quad (11)$$

where n_{pM} is the number of phones of M, n_{sM} is the number of syllables of M, i_{sM} is the position of the stressed vowel. Such a modified description function indeed involves analogically modified domain $\mathcal{D}_T$.

For practical purposes of surface prosody modelling in TTS systems, only the terminal symbols are further processed. Hence it is quite sufficient to use just a part of the vector produced by the description function (11). Moreover, significant simplification of the vector can bring benefit due to its high redundancy and low impact of some of its components.

We have experimentally selected this simplification: number of prosodic clauses of the sentence, index of the prosodic clause the prosodic word appears in, prosodeme type the prosodic word appears in, prosodeme length (measured in prosodic words), index of the prosodic word in its prosodeme, the number of syllables of the prosodic word, the number of phones of the prosodic word, index of the stressed vowel in the prosodic word. It means TTS system ARTIC assigns each prosodic word to these values. However, it is important to prove the optimality of such a simplification. This is in the scope of the future research.

3. *F*0 modelling

Let us suppose we have a suitable speech corpus (ideally the same one used for a particular speech segment database creation) with transcribed utterances, prosodic structure tags (i.e. the transcribed sentences are prosodically parsed, as introduced in the previous section) and F0 contours (e.g. acquired by electroglottograph measuring). Speech must be segmented at least on the level of prosodic words (i.e. time intervals of prosodic words must be known).

The F0 countours are segmented according to the prosodic words – this way we acquire the F0 contour of each prosodic word token (let us call such a segment a *sub-contour*). The corpus used in ARTIC consists of 5,000 sentences involving 55,655 sub-contours.

3.1. Realization function

In the process of F0 generation of a synthesised sentence the prosodic structure of the sentence is obtained first (by prosodic parsing) and then for each prosodic word its DA is determined according to (11) (or its suitable simplification respectively). Each DA is then assigned to an appropriate F0 segment using the *realization function*,

$$\mathfrak{R} : \mathcal{D}_T \rightarrow \mathcal{I} \times pot(\mathcal{C}) \quad (12)$$

where $\mathcal{I} = \{i_1, \ldots, i_l\}$ is a set of *initial conditions*, $\mathcal{C} = \{c_1, \ldots, c_m\}$ is a set of *cadences* and $pot(\mathcal{C})$ is a power set of $\mathcal{C}$. A cadence is an intonational pattern which fits into an interval of a single prosodic word. The set $\mathcal{C}$ can also be called a *cadence inventory*. Initial conditions say "where" a cadence chosen for a prosodic word starts.

Each sub-contour acquired from the corpus is decomposed into two components: (a) the initial *F*0 value of the sub-contour; (b) the rest of the sub-contour relatively to the initial value (in its multiples).

The realization function (12) also consists of two components. The first one is constructed from the corpus by linking each DA occurring in the corpus with the initial *F*0 value of the respective sub-contour occurring with this DA in the corpus. Since a particular DA is often assigned to more prosodic word tokens in the corpus, there are usually more possible initial value links. In such cases the first sub-contour with a given DA occurring in the corpus (supposing indeed arbitrary, yet constant sentence numbering) is considered – this ensures the synthesised prosodemes to be intonationally "consistent" as for the prosodic word initial conditions because the initial *F*0 values of the prosodic words within a particular synthesised prosodeme are all selected from the same sentence (otherwise it could happen that each initial condition in the synthesised prosodeme is selected from a different sentence, although with the same DA).

The set $\mathcal{C} = \{c_1, \ldots, c_m\}$ (the cadence inventory) is created by an agglomerative clustering algorithm (with various parameters – depending on a type of an experiment) applied on all *F*0 sub-contours from the corpus. Prior to this, the sub-contours are parametrised by vectors with the dimension x (e.g. by approximating each sub-contour with x equidistant points relatively to its initial value – this ensures sub-contour normalisation over time intervals and *F*0 values). The elements of $\mathcal{C}$ (i.e. cadences) are constructed as either centroids of the clusters, or there is one (or more) vector chosen from each cluster as its representant (using various methods, such as elimination of outliers according to Mahalanobis' distance).

We have experimented with various values of m (the number of cadences) ranging from 3 up to 200. Good results are achieved

for example with the number of clusters $m = 30$. In such a case the smallest cluster consists of 911 vectors (sub-contours) and the largest of 3571. The cadence inventory is created from the cluster centroids.

We say a cadence *belongs* to a particular DA provided that the sub-contour occurring in the corpus with this DA is an element of the cluster represented by the given cadence. The second part of the realization function (12) is constructed from the corpus by linking each DA occurring in the corpus with the set of all cadences belonging to this DA. Thus if we have a word w_j, then

$$\mathfrak{R}(\mathfrak{D}_T(w_j)) = \langle i_j, C_j \rangle \tag{13}$$

where $i_j \in \mathcal{I}$ is the assigned initial condition and $C_j \subseteq \mathcal{C}$, $C_j = \left\{c_{j,1}, c_{j,2}, \ldots, c_{j,l_j}\right\}$ is a set of the assigned cadences.

Now let a synthesised sentence S be given as:

$$S: \; w_1 \; w_2 \; \ldots \; w_p \tag{14}$$

The resulting generated *F*0 contour of the sentence S is formally given by the operation:

$$\arg \min_{c_{j,k} \in C_j} J\left(\mathfrak{R}(\mathfrak{D}_T(w_1)) \circ \ldots \circ \mathfrak{R}(\mathfrak{D}_T(w_p))\right) \tag{15}$$

where $j = 1 \ldots p$, $k = 1 \ldots l_j$, $\circ$ is an operation of juxtaposition (simply placing one element next to each other) and J is a criterion function selecting one cadence out of more variants for each prosodic word, as will be shown further in the text. The minimum is calculated over all assigned cadences and all prosodic words of the sentence.

3.2. Prosodic homonymy

One can easily see no corpus can offer all possible DAs and hence it is impossible to construct the realization function ideally. Thus the crucial importance for the realization function has the following *principle of exchange*:

$$\forall D_i, D_j \in \mathcal{D}_T, D_i \neq D_j : \mathfrak{R}(D_i) = \mathfrak{R}(D_j) \Leftrightarrow R(D_i, D_j) \tag{16}$$

where $R(\cdot, \cdot)$ is a *relation of indistinguishableness*. Two description arrays are in the relation of indistinguishableness provided that their different deep prosodic-semantic functions can be realized by the same functor (i.e. same surface prosodic means) – two different DAs are homonymous in terms of their surface realization and thus mutually interchangeable. Informally: the realization function is defined also for those possible DAs not occurring in the corpus; namely if a set of appropriate cadences is to be determined for a DA not occurring in the corpus, another DA which occurs in the corpus and is homonymous according to (16) is taken instead and the set of cadences and initial conditions is determined for the new DA.

A question is how to determine the essential relation $R(D_i, D_j)$ involved in (16). The best method is probably an automatic analysis of heldout corpus data – this presupposes the heldout data include DAs not occurring in the training data (i.e. factually unobserved) and the relation of indistinguishableness can be determined by a feasible generalisation of the mutual relation between the training and heldout data. This generalisation can be formalised for instance by a specific DA space metrics which allows to find a homonymous DA in terms of the minimum vector distance.

However, research in this field has not been finished yet and thus our TTS system ARTIC must now settle for a workaround in the form of performing a number of limited perturbations of the least significant (heuristically and experimentally determined) components of an unobserved DA (e.g. exact length of a prosodic word in phones, exact number of prosodic clauses in a sentence, etc.) which eventually transform the unobserved DA into such a DA that occurs in the corpus and is very likely to be still homonymous.

3.3. Criterion function

The criterion function J is responsible for choosing one final *F*0 contour from the variants proposed by the realization function. For each prosodic word of the synthesised sentence we have the initial value of its respective synthesised *F*0 sub-contour and the set of proposed cadences relatively to the initial value.

Let i_j be the initial condition of the j-th prosodic word and $C_j = \left\{c_{j,1}, c_{j,2}, \ldots, c_{j,l_j}\right\}$ the set of l_j cadences assigned to the j-th prosodic word. Each cadence $c_{j,k}$ is an x-dimensional vector of the initial value multiples, i.e. $c_{j,k} = [z_{j,k,1} \; z_{j,k,2} \; \ldots \; z_{j,k,x}]$. Given the sentence (14) of at least two words we generally use the following criterion function (or a very similar one) ensuring minimum *F*0 discontinuities between adjacent prosodic words:

$$\begin{aligned} &J\left(\mathfrak{R}(\mathfrak{D}_T(w_1)) \circ \ldots \circ \mathfrak{R}(\mathfrak{D}_T(w_p))\right) = \\ &= \textstyle\sum_{j=2}^{p} \left((\varepsilon_{j,k,1} \cdot i_j - \varepsilon_{j-1,k,x} \cdot i_{j-1})^2\right) \end{aligned} \tag{17}$$

where k indicates the k-th cadence selected from C_j for the j-th prosodic word, a "smoothed" cadence onset is $\varepsilon_{j,k,1} = \frac{1}{2}(z_{j,k,1} + z_{j,k,2})$ and analogically a "smoothed" cadence offset $\varepsilon_{j-1,k,x} = \frac{1}{2}(z_{j-1,k,x} + z_{j-1,k,x-1})$.

The first cadence is selected randomly (to enhance prosody by the natural phenomenon of randomness) from C_1 and the rest is chosen so as to minimise the function (17) according to (15) over all words and all assigned cadences (i.e. k is fixed for each j and the function J is computed, then other cadences are selected and new J computed, until J is computed for all allowed cadence combinations; eventually such a cadence sequence is chosen which gives the minimum J). The whole sentence *F*0 contour is then constructed by multiplying components of all chosen cadences with their respective initial conditions while each cadence spans the time interval of a single prosodic word.

4. Prosody quality evaluation

Each prosody generation module for a TTS system must eventually be evaluated by listening tests. Among a number of tests we have carried out particularly two of them are specially important and will be presented further in this section.

4.1. Cadence candidate number

The first version of the above described data-driven prosody model implementation in the TTS system ARTIC used only a single cadence candidate for each DA, namely the most often occurring one (in the corpus) with this particular DA. It means no criterion function J was needed (respectively, the criterion was implicitly included in the corpus analysis itself). We have carried out a listening test to evaluate the naturalness difference between the single candidate version and the multiple candidate version.

A set of sentences synthesised using both versions was prepared and 14 test respondents were asked to decide which version they perceived as more natural. The results have shown that the respondents preferred the multiple candidate version in 60% of all cases and in 20% of all cases they did not recognise any difference.

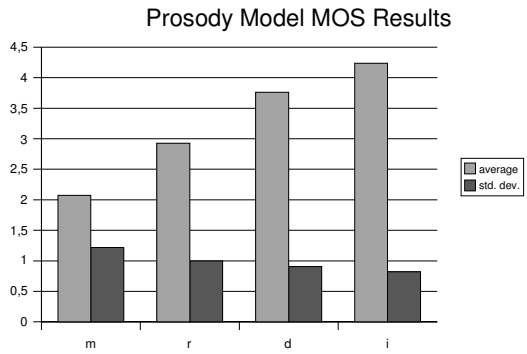

Figure 1: *The results of the MOS evaluation of the monotonous (m), rule-based (r), data-driven (d) and implanted (i) prosody of synthetic speech.*

Figure 2: *The results of the MOS evaluation of the data-driven prosody style similarity.*

4.2. Prosody naturalness evaluation

This test indirectly compares several methods of $F0$ modelling (including a rule-based method [1]) by measuring the inter-subjective criteria of the prosody quality using MOS tests (mean opinion score). This test involved 16 respondents listening to 12 different sentences, which were synthesised as follows: 3 monotonous (further denoted as *m*), 3 rule-based prosody (*r*), 3 data-driven prosody (*d*), 3 real "implanted" (denoted *i*; sentence from the corpus newly synthesised with its original $F0$). Each listener received these sentences in a random order (obviously listeners were not told which method was used to generate a particular sentence) and was asked to give each sentence a mark from the scale 1 – 5 (1 stands for worst, 5 for best) according to his/her subjective opinion. No prior "calibration" (i.e. examples of good or bad sentences) was presented to the listeners since we wanted them to express their own understanding of what (un)naturally sounding prosody is.

The results are shown in Figure 1. We can conclude this test with the following: considerably higher naturalness of the data-driven prosody model in comparison with the rule-based one is confirmed; the data-driven model is evaluated very well, i.e. not much worse than the real intonation; real intonation is surprisingly evaluated only by the mark 4 (it might point out that listeners cannot fully separate the segmental from the suprasegmental qualities of the synthetic speech, even though they are instructed to do so); the monotonous version is often evaluated as quite naturally sounding (i.e. despite of having the worst overall mark, some of the respondents gave the monotonous version even the mark 4).

4.3. Prosody style evaluation

Figure 2 displays the results of a modified listening MOS test aimed mostly at assessing how the data-driven model "copies" the prosody of the real speaker whose voice was used to record the corpus for the model training. Ten sentences from the corpus (not included in the data-driven model training) were randomly chosen and synthesised using the data-driven prosody.

The test respondents (14 persons) first listened to a sentence uttered by the real speaker and then to its synthesised version and were asked to give this sentence a mark according to the following scale: 4 – the synthesised intonation is exactly same as the real one; 3 – the synthesised intonation is significantly similar to the real one so that it is possible to recognise it comes from the same speaker (i.e. copies his/her prosody style); 2 – the synthesised intonation differs from the real one, but is still naturally sounding and appropriate for the sentence; 1 – the synthesised intonation is not appropriate for the sentence.

The results have fulfilled our expectations and show that most synthesised sentences oscillate between the score 3 and 2, which firstly means their prosody is considered to be appropriate for given sentences and secondly their prosody style is largely similar to the original one. We did not expect the score to reach the mark 1 due to the very nature of prosody itself and the nature of our data-driven model which generalises the training data and offers more appropriate intonation variants of a sentence while randomly selecting one of them.

5. Conclusions

Prosody synthesised by the proposed data-driven model has proved to be very natural and positively accepted by listeners, as underlaid by the results of the listening tests. The current research focuses mainly on experiments with a probabilistic prosodic structure parser and the theoretical background of the prosodic homonymy.

Employing the enhanced parser should significantly improve the data-driven prosody naturalness by strengthening links between sentence semantics and synthesised prosody. The prosodic homonymy relation will increase the optimality of data coverage and also hopefully contribute to linguistic understanding of the language phenomenon of prosody.

6. Acknowledgements

This research is supported by the Ministry of Education of the Czech Republic, project LC536 and project F1521/2006/G1.

7. References

[1] Matoušek J., Romportl J., Tihelka D., Tychtl Z., "Recent Improvements on ARTIC: Czech Text-to-Speech System". In: Proceedings of INTERSPEECH 2004 - ICSLP, vol. III, pp. 1933-1936. Jeju, Korea (2004).

[2] Romportl, J. and Matoušek, J., "Formal Prosodic Structures and their Application in NLP". In: Proceedings of TSD 2005, pp. 371-378. Springer-Verlag, Berlin, Heidelberg (2005).

[3] Romportl, J., "Consciousness and Causal Paradox of Emergent Systems". In: Mařík, V., Jacovkis, P., Štěpánková, O., Kléma, J. (eds.): Interdisciplinary Aspects of Human-Machine Co-existence and Co-operation, pp. 97-105. Czech Technical University, Praha (2005).

[4] Palková, Z., "Fonetika a fonologie češtiny". Karolinum, Prague (1994).