ProZed: A speech prosody analysis-by-synthesis tool for linguists.

Daniel Hirst

CNRS, *Laboratoire Parole et Langage* & Université de Provence, France
& Tongji University, Shanghai, China
daniel.hirst@lpl-aix.fr

Abstract

.
This paper describes a tool designed to allow linguists to manipulate the prosody of an utterance via a symbolic representation in order to evaluate linguistic models. Prosody is manipulated via a Praat TextGrid which allows the user to modify the rhythm and melody. Rhythm is manipulated by factoring segmental duration into three components: (i) intrinsic duration determined by phonemic identity (ii) local modifications encoded on the rhythm tier and (iii) global variations of speech rate encoded on the intonation tier. Melody is similarly determined by tonal segments on the tonal tier (= pitch accents) and on the intonation tier (= boundary tones) together with global parameters of *key* and *span* determining changes of pitch register. The TextGrid is used to generate a Manipulation object which can be used either for immediate interactive assessment of the prosody determined by the annotation, or to generate synthesised stimuli for more formal perceptual experiments.

Index Terms: speech synthesis, speech prosody, analysis by synthesis, linguistic models, rhythm, melody

1. Introduction

The interaction between linguists and engineers has always been a productive area of exchange. This is particularly evident in the area of speech prosody. The analysis by synthesis paradigm is an attractive one for linguists, since it provides an empirical solution to the problem of validating an abstract model. If the representation derived from a model can be used as input to a speech synthesis system, and if the contrasts represented in the model are correctly rendered in the synthetic speech, then the representation can be assumed to contain all the information necessary to express that contrast.

Although speech technology has become more and more accessible in recent years, it remains nonetheless true that the gap between application and users is still far too wide. This is unfortunate, since there are a great number of linguists throughout the world who are particularly interested in developing and assessing different models of prosodic structure.

Providing linguists with better tools will surely result in the availability of more and better data on a wide variety of languages, and such data will necessarily be of considerable interest to engineers working with speech technology.

In this presentation, I introduce the latest implementation of **ProZed**, a program specifically designed to allow linguists to manipulate the prosody of utterances on a symbolic level, providing an acoustic output which is directly controlled by a symbolic level of representation.

The implementation of ProZed is designed to be entirely language independent and as theory-neutral as possible, although it obviously integrates a number of non trivial principles which I have adopted over the years. It is hoped, however, that while it is never, of course, possible to be entirely theory-neutral, this software will at least prove to be *theory-friendly* in that it will be compatible with a number of different theoretical frameworks, and it may prove capable of providing evidence to allow a user to choose between various different theoretical options.

The prosody of speech can be defined for the purposes of this presentation as the explicit characterization of the length, pitch and loudness of the individual sounds which make up an utterance. Even this fairly wide definition may be found too restrictive for some, who may regret the absence of any consideration of e.g. voice quality here. In the current implementation, only the length and pitch of speech sounds are treated, since it seems likely that an efficient manipulation of loudness will require modification of the distribution of energy in the spectrum rather than simply increasing or decreasing the overall intensity of the sound. There is, of course, nothing in the ProZed framework itself which is incompatible with the representation of voice quality and this could well be integrated into the same framework, as and when algorithms for the manipulation of these characteristics of speech become more generally available.

2. The general framework.

ProZed is implemented as a plugin to the Praat software [2]. It allows the manipulation of the rhythmic and the tonal aspects of speech as defined on two specific tiers, respectively named the *rhythm* tier and the *tonal* tier. These two tiers control the short term variability of prosody. Longer term variations are controlled via a third tier named the *intonation* tier.

The speech input to the program may be natural recorded speech, the prosodic characteristics of which will then be modified by the software, or, alternatively it may be the output of a speech synthesis system with, for example, fixed (or mean) durations for each speech segment.

The current version of the program is designed as the re-synthesis step of what is planned to be a complete analysis-by-synthesis cycle. This will subsequently be directly integrated with the output of the Momel-Intsint and ProZed Rhythm analysis models which are described below as well as with the automatic alignment of phonemes and syllables as provided by the recently developed **SPPAS** tool as described in [1].

3. Using a TextGrid to modify the prosody of utterances

The annotation of the prosody of an utterance is encoded via three interval tiers. These are:

- the rhythm tier
- the tonal tier
- the intonation tier

While it is hoped that linguists will find these tiers appropriate and useful levels for modelling the rhythm and melody of speech, no assumptions are made as to the phonological units corresponding to the intervals of these tiers. *Rhythm Units*, *Tonal Units* and *Intonation Units* are consequently *defined*, respectively, as the domains of short term lengthening, short term pitch control and longer-term variation in both duration and pitch.

For different linguists, these units may correspond to different phonological entities. Thus, for example, for some linguists the Rhythm Units and/or Tonal Units may be identified with the phonological *syllable*, while for others they may correspond to larger units such as the *stress foot* or the *phonological word*.

Work with my students [9] suggests that, as originally proposed by Wiktor Jassem [11], the *Narrow Rhythm Unit* and *Anacrusis* are appropriate domains for rhythm, while the slightly larger stress foot (= Jassem's *Tonal Unit*) seems more appropriate for modelling pitch accents.

The software is designed to provides a means to implement any of these different interpretations in order to evaluate the effect of the different choice of units.

3.1. Determining segmental duration via the *Rhythm* tier.

The implementation of rhythmic characteristics in the ProZed environment makes the fairly consensual assumption that segmental duration in speech is the result of the combination of at least two factors. The first of these is the intrinsic duration of individual speech sounds. A /ʃ/ sound, for example, is intrinsically much longer than a /l/ sound.

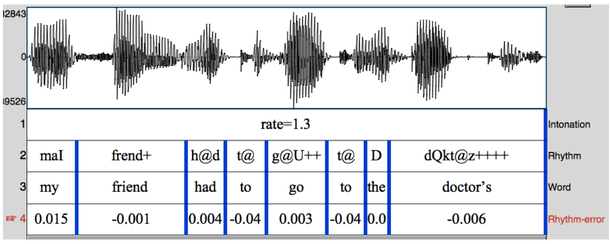
The second factor is a domain specific lengthening which in this implementation, following [9], is modelled as a scalar lengthening by an integral number of quantal units. The quantal units, by default 50ms each, are added to the sum of the intrinsic durations of the speech segments transcribed within the given rhythm unit. The resulting value is then corrected to take into account the current value of speech rate.

The formula given in [9] is:

$$\hat{d}_{ru} = (\sum_{i=1}^{m} \bar{d}_{i/p} + k * q) * t \tag{1}$$

where $\hat{d}_{ru}$ is the predicted duration of the Rhythm Unit, $\bar{d}_{i/p}$ corresponds to the mean value of the phoneme p in the corpus, q is the quantal unit of lengthening and k the scalar value of that lengthening. The final variable t, for tempo, (corresponding to $\frac{1}{rate}$), is applied to the duration of the Rhythm Unit as a whole.

To take an example, the word "go" (in figure 1), is represented on the rhythm tier as: [g@U + +].



1	rate=1.3								Intonation
2	maI	frend+	h@d	t@	g@U++	t@	D	dQkt@z++++	Rhythm
3	my	friend	had	to	go	to	the	doctor's	Word
☞ 4	0.015	-0.001	0.004	-0.04	0.003	-0.04	0.0	-0.006	Rhythm-error

Figure 1: *TextGrid for the sentence* My friend had to go to the doctor's *showing the Rhythm tier and the Word tier together with a third tier,* Rhythm-error*, generated by the program, displaying the difference between the predicted and the observed durations.*

The predicted duration of the Rhythm Unit is determined by a combination of the individual mean durations of its constituent segments, plus the lengthening factor annotated by the two plus signs.

Assuming that the individual mean durations of the phonemes /g/ and /əʊ/, as found in an external table, are, respectively 90 and 57 ms., the total duration of the *Tonal Unit* will be adjusted to 147 ms plus 100 ms of lengthening as determined by the 2 +s, i.e. a total of 247 ms, which will then be further modified by dividing by the specified speech rate factor of 1.3. The resulting predicted value of 190 ms is very close to the observed value of 187 ms.

The duration of the Rhythm Unit is manipulated linearly so that the synthesised duration is made to correspond to that determined by the symbolic representation. Thus, in the above example, the duration of the Tonal Unit containing the segment corresponding to the phonemes /gəʊ/ is globally adjusted to a duration of 190 ms. The difference between the predicted and the observed durations of each rhythm unit is calculated and displayed on a new tier (*Rhythm-error*).

The user is, of course, encouraged to experiment with different values of lengthening and speech rate in order to test various hypotheses on their interaction, as well as to experiment with different domains for the implementation of the lengthening.

In the current version of the program, there is no specific mechanism to implement final lengthening, other than by creating an ad hoc Rhythm Unit which is coextensive with the domain in which final lengthening is assumed to apply (such as the final syllable for example). This is an area in which the implementation could be improved in future versions in the light of work in progress on this type of lengthening, some preliminary results of which were reported in [7].

3.2. Determining pitch via the *Tonal* tier.

Pitch in ProZed is determined by a representation of the contour using the *INTSINT* alphabet [5]. This assumes that a pitch contour can be adequately represented by a sequence of target points, each pair of which is linked by a continuous smooth monotonic interpolation (defining a quadratic spline function).

This, in turn, assumes that the shape of a pitch-accent, for example, is entirely determined by the temporal and frequential values of the relevant target points. I have never seen a convincing example of an pitch contour which cannot be adequately modeled in this way.

The pitch height of a target is determined by the symbolic "tonal" symbol from the INTSINT alphabet which is defined either globally with respect to the speaker's current register (see below) or locally, with respect to the preceding target.

Globally, the target may be at the *top*, *middle* or *bottom* of the speakers pitch range and is consequently marked respectively as t, m or b. Locally, the pitch targets may be interpreted as being *higher*, the *same*, or *lower* than the preceding target (respectively coded as h, s or l). They may also be coded as *upstepped* or *downstepped* (u or d), corresponding to a smaller interval up from or down from the preceding target. Note that in this implementation, the INTSINT tones are represented with lower case letters rather than upper case as used in much previous work. This helps to avoid confusion with other more abstract coding schemes such as ToBI [12, 13], or the even more abstract underlying representation used in [3], both of which use some of the same symbols as INTSINT.

The actual fundamental frequency of the pitch targets is determined by the following formulas (where p is the value of the preceding target) and where pitch range is defined by the current values of two parameters key (in Hz) and $span$ (in octaves):

absolute tones:

t: $key * \sqrt{2^{span}}$

m: key

b: $key/\sqrt{2^{span}}$

relative tones:

h: $\sqrt{p * t}$

s: p

l: $\sqrt{p * b}$

interative tones:

u: $\sqrt{\sqrt{(p * t)} * b}$

d: $\sqrt{\sqrt{(p * b)} * t}$

The timing of the target points is assumed to be determined with respect to the boundaries of the corresponding Tonal Unit. In previous work (eg [4]), I suggested that this timing might be limited to a restricted inventory of positions within the Tonal Unit, such as initial, early, mid, late and final. In this implementation, I adopt a more general solution and allow in fact an arbitrary precision of alignment via the use of "dummy" targets represented by the symbol "-". Using this annotation, a tonal target X which is alone in the middle of a unit will be coded [X]. When there are more than one tonal target in a Tonal Unit, then they are assumed to be spread out evenly, so that [W X] will have one target occurring at the first quarter of the duration and one at the third quarter of the duration. This has as consequence that for two consecutive Tonal Units each containing two targets, the four targets will be all be equally spaced apart. In order to represent a target at the third quarter of the duration with no preceding target the annotation [- X] can be used. The symbol "-" is thus used to influence the timing of the other target but does not itself correspond to a pitch target.

The formula for calculating the timing of the ith target of a sequence of n targets in a Tonal Unit beginning at time $start$ and ending at time end is:

$$t = start + \frac{(2i - 1) * [end - start]}{2n} \qquad (2)$$

In practice, it is assumed that a linguist will probably make a fairly sparse use of these dummy symbols but the annotation in fact allows the specific timing of a target or targets to be coded to an arbitrary degree of precision. Thus a representation like [- - X - - Y - Z - - - -], for example, could be used to specify timing very precisely, where in this case the three targets would occur at 0.208, 0.458 and 0.625 of the duration of the interval, respectively (calculated as $2 * (i - 1)/(2 * 11)$ for i as 3, 6 and 8). The actual precision of the timing is consequently left to the user to determine. It is particularly interesting to use an annotation system which can be rendered as precise or as general as wished so that the same annotation can be used in the analysis and in the synthesis steps of the analysis-by-synthesis procedure.

3.3. Defining long term parameters with the *Intonation* tier

The short term values obtained from the Rhythm and Tonal tiers are finally modified by the long-term parameters defined on the Intonation tier. These are currently *rate* for rhythm and *key* and *span* for pitch. The three parameters are initialised with default values:

rate=1 key=150 span=1

and then any of the values can be modified for subsequent Intonation Units by simply including a specification of the value of the corresponding parameter or parameters, e.g.

rate = 1.5 span=0.8

on the Intonation tier, will make the speaking rate faster and the pitch span more reduced from that Intonation Unit on.

Each modification of a long-term value remains valid until it is modified in a later Intonation Unit. The implementation makes the assumption that changes of these parameters only occur at the onset of an Intonation Unit.

The program also allows the definition of pitch targets at the extreme ends of an Intonation Unit; using the annotation $[mb]$, for example, will place a *mid* target located at the beginning of the unit and a *bottom* target located at the end. Dummy targets can also be used here, so $[-b]$ will place only a bottom target at the end of the unit with nothing at the beginning whereas $[m-]$ will place a target at the beginning of the unit with nothing at the end. This corresponds essentially to the targets interpreted as "boundary tones" in many phonological prosodic models.

The pitch targets defined on the Tonal and Intonation tiers are output in the form of a Pitch Tier which is then converted to a quadratic spline function using the Praat function *Interpolate quadratically*. The resulting Pitch Tier is then used to replace the original Pitch via a Manipulation object, allowing the re-synthesised version of the utterance to be compared with the original sound.

4. Integrating the synthesis with the output from automatic analysis

4.1. Automatic analysis of rhythm

The model of rhythm described in [9] models segmental duration by optimising the function given above in equation 1. The output of this algorithm, an example of which is shown in Figure 2, can be used as input to the resynthesis step, as described above.

<parameter tempo=0.761><parameter quant=50> I have a problem1 with my water3 softener7. The1 water3 level1 is1 too4 high5 and the2 over-flow2 keeps2 dripping4. Could you1 a-rrange3 to send2 an engi-neer2 on Tuesday morning2 please6. It's the2 only day1 I1 can manage1 this1 week3. I'd be grateful if you could con-firm2 the a-rrangement in1 writing6.

Figure 2: *A sample passage from the Eurom1 corpus coded for duration using the automatic coding scheme described in [9]. Rhythm Units are delimited by spaces or hyphens, numbers correspond to the scalar lengthening factor k, applied to the preceding Rhythm Unit.*

The Rhythm Units are delimited by spaces or hyphens, which for the majority of English words results in a minimally economical annotation for the rhythm; a hyphen is only needed when the lexical stress occurs on a non-initial syllable, as in *engi-neer* or *a-rrange*. The scalar lengthening as described above, applies linearly to the whole preceding Rhythm Unit.

Since the lengthening factors described are applied globally to the whole Rhythm Unit, it is not necessary to have an alignment of the individual phonemes. Informal experimentation with this program suggests that this type of global lengthening of Rhythm Units is quite an efficient way of obtaining a synthetic output and that listeners are much more sensitive to differences in the duration of the Rhythm Units themselves than they are to internal differences within the Rhythm Units. This, however, is clearly an area which needs more experimental research to which it is hoped that this program may contribute.

For such precise phoneme level comparisons, the program will be interfaced with the recent SPASS algorithm described in [1].

4.2. Automatic analysis of pitch

The output of the Momel and Intsint algorithms, as described in [10, 5] and implemented in [6] can be directly used as input to the re-synthesis module as described above. The temporal alignment of the targets currently needs to be determined manually with respect to the boundaries of the Tonal Units, but this step will also be automated in future implementations of the program.

5. Conclusions

The ProZed plugin is designed as a tool to enable linguists to manipulate the rhythmic and tonal properties of an utterance by means of a symbolic representation, in order to evaluate the appropriateness of different phonological models of pitch and rhythm. It can be used either for immediate interactive experimentation with prosodic annotation, or to generate synthetic stimuli for more formal perceptual experiments. The plugin is freely downloadable from the *Speech and Language Data Repository*:

http://sldr.org/sldr000778/en.

This software, in conjunction with the algorithms for the automatic analysis of speech prosody described in [1, 8, 9], aims to provide a complete analysis by synthesis environment, which, I have argued, is crucial for the development and testing of empirical models of speech prosody.

6. References

[1] Bigi, Brigitte and Hirst, Daniel. SPPAS: a tool for the automatic analysis of speech prosody. *6th International conference on Speech Prosody, Shanghai, PRC.* [This volume].

[2] Boersma, Paul and Weenink, David. *Praat: a system for doing phonetics by computer.* Available from http://www.praat.org. 1992-2010.

[3] Hirst, Daniel. Intonation in British English. In Daniel Hirst & A. Di Cristo (eds.) 1998. *Intonation Systems. A Survey of Twenty Languages*. Cambridge: Cambridge University Press. 56-7 1998.

[4] Hirst, Daniel. The symbolic coding of duration and alignment. An extension to the INTSINT system. *Proceedings Eurospeech '99*. Budapest, September 1999.

[5] Hirst, Daniel. Form and function in the representation of speech prosody. in K.Hirose, D.J.Hirst, Y.Sagisaka (eds) *Quantitative prosody modeling for natural speech description and generation* (= *Speech Communication* 46 (3-4)), 334-347. 2005.

[6] Hirst, Daniel. A Praat plugin for MOMEL and INTSINT with improved algorithms for modelling and coding intonation. In *Proceedings of the 16th International Congress of Phonetic Sciences*. Saarbrücken. Germany. 1233-1236. 2007.

[7] Hirst, Daniel. The rhythm of text and the rhythm of utterances: from metrics to models. *Proceedings of Interspeech 2009*, Brighton. 1519–1523, 2009.

[8] Hirst, Daniel. The analysis by synthesis of speech melody: from data to models. *Journal of Speech Sciences*, 1(1):55–83, 2011.

[9] Hirst, Daniel; Bouzon, Caroline and Auran, Cyril. Analysis by synthesis of British English speech rhythm: from data to models. In G. Fant, F. Hiroya, and S. Jiaxuan, editors, *Frontiers in Phonetics and Speech Science. A Festschrift for Professor Wu Zongji's 100th Birthday.*, 251–262. Commercial Press, Beijing, Peoples Republic of China, 2009.

[10] Hirst, Daniel and Espesser, Robert. Automatic modelling of fundamental frequency using a quadratic spline function. *Travaux de l'Institut de Phonétique d'Aix* 15. 71-85. 1993.

[11] Jassem, Wiktor. *Intonation of Conversational English: (educated Southern British)*. Nakl. Wroclawskiego Tow. Naukowego; skl. gl.: Dom Ksiazki, 1952. [available from the *Speech and Language Data Repository* http://sldr.org/sldr000777/en]

[12] Pierrehumbert, Janet B. *The Phonology and Phonetics of English Intonation.* Massachusettes Institute of Technology. Ph.D. dissertation [Published in 1987 by Indiana University Linguistics Club, Bloomington]. 1980.

[13] Silverman, Kim; Beckman, Mary E.; Pitrelli, John; Ostendorf, Mari; Wightman, Colin; Price, Patti; Pierrehumbert, Janet B.; Hirschberg, Julia. ToBI: a standard for labelling English prosody. In *Proceedings, Second International Conference on Spoken Language Processing* 2. Banff. Canada. October 12-16. 867-870. 1992.