Development of a BOSS unit selection module for tone languages

Arne Bachmann, Stefan Breuer

Institute of Communication Sciences (IfK)
University of Bonn, Germany
arne.bachmann AT web.de, breuer AT ifk.uni-bonn.de

Abstract

The Bonn Open Synthesis System (BOSS) is a toolkit for the efficient development of speech synthesis applications. To facilitate adaptation to tone languages, we added support for tone contour quantization and prediction. Now it is possible to integrate syllable and word tone templates into the system and predict as well as select them efficiently. The simple model presented here is trained automatically and works independently of the morphophonemic rules specific to a certain tone language. Its feasibility is exemplified for the African language *Ibibio*.

1. Introduction

1.1. BOSS

The *Bonn Open Synthesis System* BOSS [1] is a research and development platform originally written for unit selection-oriented speech synthesis, but also applicable to other approaches [2]. Its building blocks are reusable libraries and language modules (German, Polish, Ibibio) in C/C++. BOSS also provides tools for creating and optimizing corpora. The system communicates and stores data using XML files and their DOM representations; runtime access to corpus data is optimized for speed by use of MySQL [3] databases (DB). BOSS is a network-enabled application. Communication between synthesis server and clients works over a simple protocol that hosts XML and audio data. Synthesis can be executed from the command line or by a Java GUI client. BOSS provides a bootstrap install mechanism and thus can be installed and run on Unix-based platforms. German online-synthesis is available at the BOSS website [1]. As a platform intended for research, BOSS is not optimized for limited resources, although many ideas for optimization are conceivable and waiting for implementation. Feel free to contribute to the BOSS project.

1.2. Objective

Our first aim was to write a BOSS intonation/unit selection module for the African tone language *Ibibio*. In cooperation with Prof. Urua (University of Uyo, Uyo, Nigeria), the fundamentals of the module were planned and worked out in [4].

The main objectives for a general tone language intonation module are adaptability, extensibility, simplicity, ease and speed of development, run-time speed, universality and knowledge gain through machine learning (ML). Since tone languages like Ibibio exhibit intriguingly complex intonation, e. g. may combine phenomena such as declination, downdrift, downsteps, final fall and tone assimilations, it is very hard to derive rules for a rule-based intonation synthesis manually. Some rough rules for Ibibio can be found in the literature, e. g. lowering of the topline by 30 Hz after downsteps and final fall of about 10 Hz [5], but there is no integrated model to describe the interactions between the various influences on the suprasegmental structure of the language. To avoid the tedious search for uncertain regularities, we leave it to the machine to learn the patterns of intonation and therefore gain a reusable and easily retrainable system.

1.3. Ibibio

Ibibio is one of over 1500 Niger-Congo languages and is spoken in the southwestern part of Nigeria by about 5 million people. The language has three tonemes (high **H**, low **L** and downstepped high **D**), plus two non-contrasting surface contour tones (rising **R** and falling **F**) [6]. Usually, tones are not represented in orthography. Ibibio shows interesting tonal features: Tones are lexically and grammatically distinctive. There are complex morphophonemic word tone templates (cf. [5]):

Ibibio	**English**
sé	look
áà-sèè-hè	one who looks
áà-!ké-séé-hé	one who looked
áà-!dî-sé	one who will look
nò̥	give
áà-nò̥ò̥-hò̥	one who gives
áà-!ké-nò̥ò̥-hó̥	one who gave
áà-dî-nò̥	one who will give

In this notation, **!** is a downstep, the subring shows the presence of a deleted underlying (floating) tone. There is also downdrift [7]. This means that like consecutive tones have the same fundamental frequencies; they may need some time to reach the target frequency, though, a phenomenon called *start-up effect* [5].

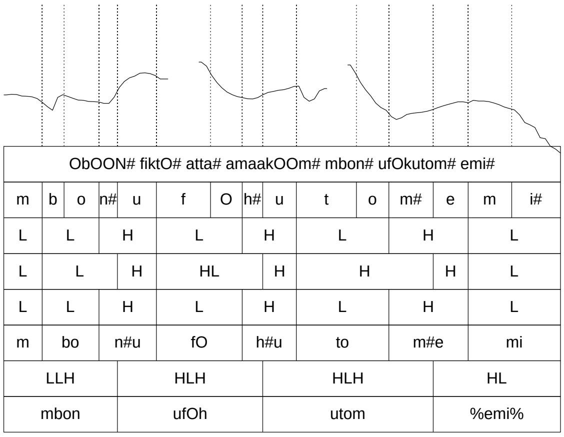

Figure 1: *Sample corpus sentence. The annotation tiers shown are (from top to bottom): Sentence transcription, phones, surface tones, underlying tones, mapping of tier 3 to single letter tone symbols, syllables, word tones, words.*

For the research presented in this paper, we used a small annotated Ibibio corpus of 94 sentences spoken by Prof. Urua. An excerpt of the corpus is shown below.

121: /ana ekop nsɔŋidem odo ɔjɔhɔ ɔjɔhɔ ke ifuuro uwem/
122: /ekæriku odo akwa owo eto/
123: /mmɔɔŋ idım ukana amaasideŋŋe/

2. Methods and models

2.1. The linear approach

As for most languages we do not know the optimal intonational decomposition in advance, we must leave it to the ML to learn it. Our model predicts a sequence of syllable tone contours on the basis of the symbolic description of a target utterance's tonal surface structure. We simply feed it alongside all other available parameters such as number of preceding hightones or downsteps into the ML, thus not anticipating any language-specific features or assuming erroneous regularities. To adapt our module to a new language, the only necessary change to the ML is the choice of features, derived from recommendations in the literature. Other reasons for preferring a linear approach over superposition are:

- Ease of data extraction: Obtaining the observable surface intonation contour is straightforward
- No global component necessary (e. g. phrase commands)
- The tone-bearing unit (TBU) is usually a syllable, so complexity is reduced by concatenating syllable contours

One problem arising from the linear tone sequence model is the possibility of F_0 discontinuities at concatenation boundaries. This problem seems neglectable, because a well-trained prediction module should produce only small differences at syllable boundaries which can be smoothed easily in unit selection and signal manipulation.

2.2. F_0-Stylization

For automatic extraction of fundamental frequency contours, we used ESPS' *get_f0* [8] as well as *Praat* [9]. For the application to our Ibibio corpus, we faced the problem that syllable boundaries were not annotated. Thus we had to resort to the boundaries marked for surface tones.

In contrast to modeling intonational events known from accent languages e. g. by using the Tilt model [10], we use quartic polynomials to stylize syllable contours. Our method does not require any suprasegmental markup and therefore rather resembles the PaIntE model [11]. One advantage of using polynomial regression is ease of implementation. On the downside, there may be unsolvable systems of equations due to a lack of data points, e. g. in voiceless sections or where the extraction algorithm calculated F_0 values out of a sensible range.

To determine the optimal polynomial order, we tested the quality of stylization for different regression settings on the Ibibio corpus. To this end, we stylized and resynthesized the intonation contours using original syllable durations. The differences in the extracted F_0 values between original and polynomial contours were measured by means of a Perl script. Contour accuracy is shown in table 1. The values represent root mean square error (RMSE) and mean absolute errors (MAE) in Hertz plus Pearson's correlation coefficient r for different polynomial orders. We also experimented with stylization on a logarithmic scale, but results were slightly less satisfactory.

In the end, stylization was done using polynomials of fourth order, as this gave the best ratio between approximation quality and the number of solvable equation systems and thus the number of syllables that could be used for training. The polynomials are stored using both their coefficients and a data point representation. The latter is derived by computing the polynomial values for the left and right syllable boundaries, the middle, and two points between the middle position and the borders. This way, we get tone contour descriptions that are independent of the syllable durations. We don't apply an F_0 normalization to the contour shapes, because the absolute data values may serve to distinguish different intonational functions.

2.3. Data reduction

We use a simple vector quantization (VQ) approach to reduce the syllable contour data. This serves two purposes: firstly, to create a reasonably concise amount of distinct syllable contours for machine learning, and secondly to gain knowledge over the most common syllable contours and their linguistic/phonological distribution. The data reduction method used in our module is the well-known LBG algorithm [12]. To reduce the set of observed syllable contours, their polynomial functions are taken to form a vector space by using five equidistant data points. Thus, each syllable can be represented by a quartic polynomial. We don't use the polynomial coefficients at this stage, as their values have different dimensions and cannot be compared by simple distance measures as necessary in the VQ.

The LBG algorithm successively divides the vector space into halves. The resulting 2^N prototypes, collected in a *codebook*, represent an optimal partitioning of all data points (read: syllable contours) in the vector space. As can be seen in figure 2, we chose a codebook size of 64 entries, which was the best choice for the current small size of the corpus.

Afterwards, the codebook's prototypes themselves were vector quantized to get a set of superclasses — a further layer of abstraction to be used as a fallback in unit selection whenever there is no unit available for a certain prototype. For our corpus, the best combination of codebook and codebook classes in terms of distortion minimization was achieved for 64/16 codevectors, as can be seen in table 2. Re-quantizing the existing codebook, as if it were a set of original contour shapes attempts to reduce data that is already optimally distributed in the vector space. Additionally, the original frequency distribution (number of data vectors per codevector) is neglected. This approach can be improved. One possibility would be to create a smaller codebook from the original data and to assign the original data points to these classes. When looking at the produced protoypical syllables shapes of the codebook in figure 2, one can observe several properties of the algorithm: The overall fundamental frequency curve rises with the number of codevectors and 2^N neighbouring codevectors share some features, e. g. rising or falling shape, while varying in others.

The VQ automatically tries to make its codevectors represent the data vectors best, so we can assume that a fairly large codebook represents the most frequent tone contours of a language. The codebook provides the essential interface between surface acoustic and surface symbolic-phonetic information and with that, the phonological categories[1].

Order	RMSE [Hz]	MAE [Hz]	r
1	10.55	6.74	0.964
2	6.98	4.26	0.985
3	5.47	3.17	0.991
4	**4.53**	**2.49**	**0.994**
5	4.02	2.14	0.995

Table 1: *F_0-stylization accuracy for various polynomial orders. The order used is printed in bold letters.*

2.4. Prediction

Our choice for a prediction method started with the following considerations:

- There should be tools available for training
- Robust creation and prediction
- Low implementation and integration cost for BOSS
- Human-understandable ML knowledge gain

Thus, neural networks (NN) and support vector machines (SVM) were discarded in favor of classification and regression trees (CART) [13].

Advantages of CARTs include:

- Very fast execution and low memory usage in working phase (binary trees)
- Already implemented in BOSS as a parser for LISP-like decision tree files
- Good tools available (wagon [14]), simple setup and training
- No black box: Human-comprehensible and extensible decisions in a simple tree structure. Potential linguistic knowledge gain concerning tonal phenomena

In the German BOSS module, regression trees are already used for phone duration prediction. We extended the source code to predict classes in addition to mean/standard deviation pairs on the tree leaves in order to be able to use the implementation both for tone contour and duration prediction in the Ibibio module. By using CARTs, it is possible to recognize superpositions in the tree structure as similar returning decisions in different branches. We should thus be able to detect the individual influences of the input parameters on the resulting tonal contours. Disadvantages of the CART include: The importance of single training parameters may vary strongly upon but small changes to the DB. Secondly, like all data-driven machine learning methods, the availability of large amounts of reliable data is essential for successful training.

[1] Presuming the relation between phonology and symbolic surface structure is known.

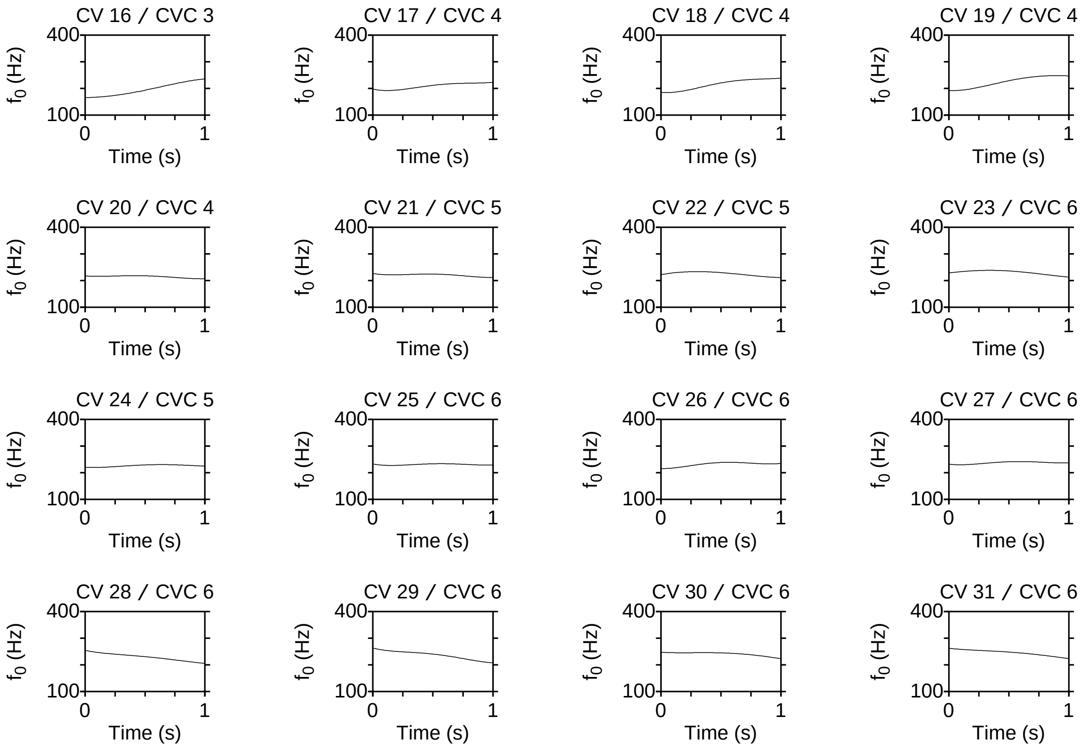

Figure 2: *Vector quantization: Codebook excerpt for codevectors 16. . .31 out of 64*

<table>
<tr><th>Order</th><th>Syllables</th><th>Size C_1</th><th>Distortion</th><th>SNR</th><th>Size C_2</th><th>Distortion</th><th>SNR</th></tr>
<tr><td>3</td><td>2333</td><td rowspan="6">64</td><td>230.31</td><td>-23.62</td><td rowspan="3">8</td><td>568.68</td><td>-27.55</td></tr>
<tr><td>4</td><td>2322</td><td>224.03</td><td>-23.50</td><td>531.99</td><td>-27.26</td></tr>
<tr><td>5</td><td>2067</td><td>240.54</td><td>-23.81</td><td>5656.15</td><td>-37.52</td></tr>
<tr><td>3</td><td>2333</td><td>230.31</td><td>-23.62</td><td rowspan="6">16</td><td>314.46</td><td>-24.98</td></tr>
<tr><td>4</td><td>2322</td><td>224.03</td><td>-23.50</td><td>290.29</td><td>-24.63</td></tr>
<tr><td>5</td><td>2067</td><td>240.54</td><td>-23.81</td><td>4836.02</td><td>-36.84</td></tr>
<tr><td>3</td><td>2333</td><td rowspan="6">128</td><td>156.82</td><td>-21.95</td><td>447.28</td><td>-26.51</td></tr>
<tr><td>4</td><td>2322</td><td>156.35</td><td>-21.94</td><td>389.75</td><td>-25.91</td></tr>
<tr><td>5</td><td>2067</td><td>170.50</td><td>-22.32</td><td>3762.66</td><td>-35.75</td></tr>
<tr><td>3</td><td>2333</td><td>156.82</td><td>-21.95</td><td rowspan="3">32</td><td>309.64</td><td>-24.91</td></tr>
<tr><td>4</td><td>2322</td><td>156.35</td><td>-21.94</td><td>258.54</td><td>-24.13</td></tr>
<tr><td>5</td><td>2067</td><td>170.50</td><td>-22.32</td><td>2744.57</td><td>-34.38</td></tr>
</table>

Table 2: *Vector quantization: Comparison of different codebook sizes and polynomial orders, smoothing of all voiceless syllable parts. Values given are overall distortion in data fitting and signal-to-noise ratio (SNR: $-10\log_{10} dist$). The best codebook size combination is printed in bold letters.*

A typical training set of 84 sentences for CART constructed from our Ibibio corpus would leave only about 4'30" of speech, not counting pauses. This is the amount left after removing ten sentences for testing purposes, which is clearly not enough to demonstrate the full potential of our approach. The results presented here should thus be seen as a preliminary estimation. Applying the CART for tone contour prediction to the test set rendered results ranging from 38.55 - 59.04 %, the large interval between the outcomes already indicating a sparsity problem. Duration prediction results were slightly better, but still unsatisfactory for the same reason. Table 3 lists the five most important parameters in codevector and duration prediction trees, taken from sample training set no. 7. The parameters *sylphrase wordphrase* represent the position of the respective unit in the phrase, *sylsphrase* and *wordsphrase* the number of these units it is comprised of. *Sylstruc* encodes the syllable type[2]. The left and right tonal context of each syllable was captured by *ltone4...ltone1* and *rtone1...rtone4*. Parameter *d* is the number of preceding downsteps in the phrase, and *firstcons* stands for the first consonant of the syllable. The distances to left and right phrase boundaries are given by *bodil* and *bodir*. Other features used for training are *r* and *f* for the number of preceding L-H and H-L tone shifts, respectively. For numbered features we also added categorical versions with the possible values initial, medial, final and single. The features used for prediction were collected from recommendations in the literature; an explanation of all features is given in [4]. After training, the

Contour classification		Duration regression	
Feature	**% correct**	**Feature**	**% correct**
sylphrase	62.3	sylstruc	80.7
wordphrase	64.5	rtone	85.7
sylstruc	66.4	sylphrase	87.4
rtone3	67.4	firstcons	88.3
d	68.3	bodir	88.7

Table 3: *Most important five prediction features for tone template and duration CARTs and their cumulative prediction accuracy.*

decision structure of the tree was analyzed. Especially the role of the number of downsteps and of downdrift was inspected, but the impact of downsteps predicted in the literature was not transparent in the CART. Since *sylphrase* was the dominant decision feature in most trained trees, we would rather assume a declination component for the current data. More data has to be segmented and annotated for further investigation of the role of *r*, *f* and *d*.

[2]The symbols C, V and N were used to represent consonants, vowels and nasals respectively. The latter were included to account for the special importance of nasals in Ibibio.

With respect to the different tonal shapes, only one valley shape was found in the codebook. Thus, a more restricted parametrical representation might also have worked.

2.5. Unit selection

BOSS employs a stepwise reduction of unit search criteria called preselection to reduce the number of database lookups. Thus, if no perfect fitting unit can be found — judging from the symbolic description only — the context is widened and other possible, but less narrowly defined, units come into selection focus.

We introduced two new cost functions to the Ibibio module: To compare the syllable tone contours, the data points from the codebook and those found in the corpus units are compared via RMSE. On the phone level, a categorical measure for the position inside the syllable was introduced with initial, medial, final and single as possible values. For mean syllable F_0 unit and transition costs, the standard BOSS approach is used.

Determining the weighting (or cost) factors for the different unit selection cost functions is a non-trivial problem. In our approach, we normalized all cost functions by their corpus mean value and weighted them in same parts.

A critical problem was the small corpus size: Even after widening the search focus maximally, for some test sentences no fitting syllables or even single phones were found in the database. This calls strongly for a bigger corpus. Additionally, the Ibibio module was originally designed only for syllable-based synthesis, so that phone synthesis represents an unsatisfactory solution. This stems from the fact that we predict syllable tones, and therefore it is hard to tell if a phone fits a given syllable contour. The forementioned phone cost term is one method to remedy this.

2.6. Signal manipulation

Until now, no signal manipulation has been implemented. There are two reasons: Corpus synthesis should in principle work without manipulation (and it does) and development time was restricted to six months in [4]. In principle, BOSS supports PSOLA manipulation, but the modules expect F0 contours as input which would have required an additional transformation function for codevectors. While the general algorithm for recreating a polynomial shape from the codevectors can be found in the BOSS-IBB documentation [15], it was not implemented in this first version of the Ibibio package modules.

3. Discussion

We have shown a syllable-based tone contour codebook synthesis with CART ML to be feasible. We believe that our model should be applicable to other tone languages and our prototypical implementation for Ibibio could serve as a template for the creation of other language adaptations. So far, we have presented some evaluation results on the accuracy of polynomial fitting and vector quantization. With only the small amount of Ibibio data at hand, meaningful subjective listening tests with native Ibibio speakers could not be conducted. Data sparsity affected not only the reliability of the CART trees but also the number of units to choose from for synthesis. Thus, the next step will have to be the creation of a much larger corpus to synthesize from and retraining of the CART and CBs, as well as testing the method on other languages. Criteria to examine in listening tests based on the new data could pleasantness, naturalness, intelligibility and overall intonation.

Some of the technical work under way is the creation of an independent reference module as a starting point for other language modules. This is planned to be done for BOSS-IBB V 0.2. Other language adaptions waiting for realization are Yoruba and Chinese. To test the applicability to accent languages, the method shall be evaluated for German as well.

Other future plans include the improvement of tone template classes and a closer examination of the phonological role of downsteps and downdrift in Ibibio.

4. Acknowledgements

We would like to thank all the people who supported us with their knowledge and by providing additional source code: Axel Bußmann (Osnabrück / Germany), Moses Ekpenyong (Uyo / Nigeria), Prof. Dafydd Gibbon (Bielefeld / Germany), Prof. Wolfgang Hess (Bonn / Germany), Nam Phamdo (Laurel / United States) and Prof. Eno-Abasi Urua (Uyo / Nigeria).

5. References

[1] [Online]. Available: http://www.ikp.uni-bonn.de/dt/forsch/phonetik/boss/

[2] P. Birkholz, I. Steiner, and S. Breuer, “Control Concepts for Articulatory Speech Synthesis,” *This conference*, 2007.

[3] “MySQL 5.0 Reference Manual,” 2006. [Online]. Available: http://www.mysql.org

[4] A. Bachmann, “Ein quantitatives Tonmodell für Ibibio. Entwicklung eines Prädiktionsmoduls für das BOSS-Sprachsynthesesystem,” Master’s thesis, University of Bonn, Aug 2006.

[5] E.-A. E. Urua, “The tone system of Ibibio,” in *Typology of African Prosodic Systems Workshop*, may 2001.

[6] E.-A. E. Urua, *Ibibio*, ser. Journal of the international phonetic association. International phonetic association, 2004, no. 34/1, ch. Ibibio, pp. 105–109.

[7] B. Connell, “Downdrift, Downstep and Declination,” in *Typology of African Prosodic Systems Workshop*. Bielefeld University, may 2001.

[8] [Online]. Available: http://computing.ee.ethz.ch/sepp/esps-5.3.1-vj/

[9] P. Boersma and D. Weenink, “Praat: doing phonetics by computer (Version 4.5.19) [Computer program],” 2005, retrieved April 15, 2007. [Online]. Available: http://www.praat.org

[10] K. Dusterhoff and A. W. Black, “Generating F0 contours for speech synthesis using the tilt intonation theory,” in *Proceedings of the 1997 ESCA Workshop on intonation*, Athens, Greece, 1997.

[11] G. Möhler and A. Conkie, “Parametric modeling of intonation using vector quantization,” 1998.

[12] Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” in *IEEE transactions on communications*, vol. 28, jan 1980, pp. 84–95.

[13] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York: Chapman and Hall (Wadsworth, Inc.), 1984.

[14] P. Taylor *et al.*, *Edinburgh Speech Tools Library - System Documentation Edition 1.2, for 1.2.0*, June 15 1999.

[15] A. Bachmann, *BOSS-IBB. Speech Synthesis Module Documentation for the BOSS Ibibio module*, University of Bonn, Apr 2007, revision 4.