DURATION MODELING IN A RESTRICTED-DOMAIN FEMALE-VOICE SYNTHESIS IN SPANISH USING NEURAL NETWORKS

R. Córdoba, J.M. Montero, J. Gutierrez-Arriola, J.M. Pardo
Grupo de Tecnología del Habla. Departamento de Ingeniería Electrónica. Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040 Madrid, Spain
cordoba@die.upm.es
http://www-gth.die.upm.es

ABSTRACT

The objective of this paper is the accurate prediction of segmental duration in a Spanish text-to-speech system. There are many parameters that affect duration, but not all of them are always relevant. We present a complete environment in which to decide which parameters are more relevant and the best way to code them. This work is the continuation of [1], where all efforts were dedicated to an unrestricted-domain database for a male voice. In this case, we are considering a female voice in a restricted-domain environment. This restricted-domain offers several advantages to the modeling: the variation in the different patterns is reduced, and so most of the decisions we have made about the parameters are now based in more significant results. So, the conclusions that we present now show clearly which parameters are best. The system is based in a neural network absolutely configurable.

Keywords: prosody, duration, text-to-speech, neural networks, parameter selection and coding, restricted-domain synthesis.

1. INTRODUCTION

The primary goal of this study was to adapt our automatic system used to model duration for a Spanish unrestricted-domain text-to-speech system to a new female voice in a restricted-domain.

At the same time, we wanted the system to be very flexible and we succeeded, because we have been able to adapt it easily to a new context with very little work in the configuration of the system.

Many studies have been successfully carried out lately in the field of automatic estimation of a duration model, using different techniques and input parameters to obtain the model. For example, Eloquens [4] is a system for Italian using decision trees. The parameters considered in this system are: phoneme identity, stress, window of phonemes, and characteristics of higher order units (syllable, word, and phrase).

In all the systems, regardless of the technique used for the modeling, it is crucial to find the parameters that are most significant for duration modeling. So we can take advantage of previous studies dedicated to duration modeling, but using other techniques. For example, the technique using a sum-of-products model from ([5][6]) used the following parameters: phoneme identity, stress, position in the word (initial, medium, and final), word length in syllables (1, 2, 3, or > 3) and diphthong.

Neural networks have previously been used with success. In [3] a syllable duration modeling is shown, in which the neural network is used to predict the standard deviation of the duration (called the "syllable elasticity factor"). The parameters used are the breaks, the stress, and the information on the composition of the syllable.

We analyze new alternatives for the parameters used and its codification as inputs to the neural network.

A domain-specific application does not require so many sentence structures, but there are many words embedded in them. Although the delivered messages are syntactically constraint, the vocabulary size is potentially huge. A message is typically a sentence with two different parts: one of them, that is fixed, is a template for the other, which is composed of one or more slots (Variable Fields) containing the relevant information that the user is looking for in the message.

Current prosodic patterns are judged as too monotonous to allow a great diversity of services. But in restricted-domain applications and by mixing female natural speech and diphone-concatenation synthesis (from the same speaker), we can provide high quality services. However, this contrast of human and artificial voices forces synthetic speech to be as close as possible to natural voice. The speech synthesis obtained tends to mimic the natural prosody exhibited by the speaker [7].

We have used first about 75% of the database for training and 25% for testing. The division has been made according to phrases, not to phonemes, trying to be as homogeneous as possible.

2. DATABASE FOR TWO RESTRICTED DOMAINS

These database has been used too for F0 modeling and a initial version of the duration model. In [2] we can see a thorough description of the database generation.

We extracted a set of 19 Carrier Sentences (CS) from two real services in banking and traffic information domains, provided by the IVR company that made the design of the dialogue. They contained 24 Variable Fields (VF). As each VF conveys the most important information in the sentence, and for further restricting the prosody, the professional speaker had to utter each VF between 2 compulsory pauses.

We can classify the sentences into 3 classes:

0-7803-7041-4/01/$10.00 ©2001 IEEE

- *Proper Names* (PN): 9 CS with 11 VF, that include surnames (both compound and simple ones), cities and villages, and mountain roads.
- *Questions* (Q): 4 interrogative CS with 4 VF containing bank-related information: currency, cheque status, etc.
- *Noun Phrases* (NP): 6 CS with 9 VF, also regarding to banking information: accounts, credit cards, names and types of financial transactions, banks. We include these later items in the NP class because they are syntactically related to NP, as in Caja de Ahorros y Monte de Piedad de ... or in Banco de Crédito Local de ..., where the names of these banks include a typical Noun Phrase structure with one or more Prepositional Phrases.

There are total of 1735 phrases, 3594 words, 6551 syllables, and 20089 phonemes (balanced between the three classes).

The recorded database was then phonetically labeled in a semiautomatic way. We used a continuous speech recognition system with HMM models, and manually revised the outcome using a GUI adjusting program.

3. DESCRIPTION

As we did in the previous article [1], we have focused our work in the following problems:

1) Which topology should we use for our network?
2) Which is the best way to code the parameters?
3) Which are the best parameters that we should use?

3.1 Topology of the neural network

We have used a multilayer perceptron (MLP) and the sigmoid as activation function.

Our basic unit is the phoneme. For each phoneme, we have a series of coded parameters. There is one output in our network: the duration of the phoneme.

As we obtained in [1], we always get better results using just one hidden layer. The best procedure to know the optimum number of neurons that the net should have is to increase the number of neurons one by one and observe the test results (training results are not significant for our problem, as they always improve using more neurons), and stop when there is an overtrain symptom (decrease for the test set).

In this restricted-domain system we had the option to use a single network for the 3 classes of sentences or 3 different networks for each class. Using the best configuration of parameters of [1] we compared both approaches. The second option (3 networks) improved the results in 6% (test set) and 8% (training set), so we decided to use 3 different networks in our experiments.

3.2 Codification of the parameters

We have considered different ways present the parameters to the neural network, i.e., the way they are coded, as we have different kind of parameters.

1) **Binary coding**: this is the standard coding for true/false parameters.
2) **One-of-n**. We use n neurons and only one of them is active, the one that corresponds to the class or category.

In ordinal values there is a relationship of order between the different values. For example, the position of a unit inside a higher-order unit. For these values we have considered three codifications:

3) **Porcentual transformation**: divide the current value by the maximum value to obtain a percentage. We have a floating-point value as input.
4) **Thermometer**: divide all the possible values in different classes (intervals). We activate all the neurons until we get to the current class and leave the remaining neurons inactive.
5) **Z-Score mapping**: apply a normalization to the floating-point value that takes into account the average and the standard deviation of the variable. It is a good codification for very variable parameters.

3.3 Network evaluation

To evaluate and compare the networks we have considered different metrics for the error (difference between the prediction from the network and the optimum value). The most important metric is the MSE, or the RMS, which is equal to sqrt(MSE). They are both more reliable than the average absolute error.

To make our comparisons it is better to use an adimensional metric, because it will be independent of the way we code the duration. We decided to use the following one because it does not have an offset:

$$\mathrm{Re}\,lative\ RMS\ error(2) = \frac{RMS}{\sqrt{\sum (t_i - \bar{t})^2}}$$

where t_i are the optimum values.

3.4 Parameters to be used

As we found out in [1], it is too difficult to decide which parameters are relevant and the best way to code them using a very big network with many parameters, because the differences in performance are too small and not always significant. So, we have used a base experiment using only the phoneme identity and the stress, which are the most relevant ones without doubt. Then, we have added the different parameters one by one to see the significance of each of them. Obviously, we considered the best options we obtained in [1].

3.4.1 Modeling of the duration

We obtained in [1] that the duration should be normalized. We found that it is better to normalize it by the duration of the phrase; this way the system is less affected by changes of speed in the database recordings. After the normalization, we use the logarithm of the standard deviation and a Z-Score codification.

3.4.2 Phoneme identity

We have considered a set of 33 phonemes for Spanish and used a one-of-n coding.

Contextual phonemes: we have used the phonemes that are to the right and to the left of the current one. The number of inputs is too high, so we had to use 13 clusters of similar phonemes: we classified the left and right context phonemes in these classes. This way we reduce the inputs to 59 (33+13+13). Our reference system uses this 3-values window. In Table 1 we show the results with a 5-values window (experiment 2) which show an improvement (a 7-values window gives a worse performance, probably because there are too many parameters).

3.4.3 The stress

The effect of this parameter is always important. The coding is binary: the phoneme can have stress or not. We have obtained better results using a window of five stress values to include contextual information. See Table 1.

3.4.4 Position in phrase in relation to stress

We code each syllable in 5 possible classes:

- beginning of phrase till first stress
- first stress
- after first stress till last stress
- last stress
- from last stress till ending of phrase.

As we can see in Table 1, the results are good. Using less than 5 classes the results were worse (and are not shown). This is different from what we observed in the unrestricted-domain system, where the improvement was lower and the best option was to use only 2 classes.

3.4.5 Binary parameters

We show below another binary parameters that have been considered in our experiments. All of them have shown an improvement over the reference experiment (see Table 1), which is more significant than in the unrestricted-domain system.

- Diphthong.
- Syllable beginning with vocal.
- Phoneme in a function word.

3.4.6 Type of phrase

We have five different types of phrases [2], but all experiments that considered this parameter with different codifications showed worse results. The reason is that we are using three different networks and the distribution of types of phrase is very unbalanced. So, we will not use it in the final network.

3.4.7 Position in the phrase

We decided in [1] that the most coherent alternatives for position parameters are: position of the phoneme in the syllable, the syllable in the word, and the word in the phrase. These are the steps followed for the codification:

- Normalize the value of position by the total in the higher-order unit – we obtain a floating point value between 0 and 1.
- This value is coded using 3 classes. The intervals that define these classes are computed automatically looking for uniform distributions.
- We then use a thermometer-type codification (2 neurons, always the number of classes - 1).

The results are shown in Table 1. The best one is 'position of the word in the phrase', one conclusion that we did not obtain in the unrestricted-domain system, where all parameters related to phrase provided worse results. The reason is that the range of values is much more uniform in the restricted-domain system.

3.4.8 Number of units in the phrase

In a similar way than for positions, we have considered the number of phonemes in the syllable; syllables in the word; and words in the phrase. We followed these steps for their codification:

- Normalize the value of position by the maximum value – we obtain a floating point value between 0 and 1.
- Apply Z-score (using average and standard deviation): we can restrict at our will the operating range of the parameter.

The number of words in the phrase is the best parameter. Again, this a difference with the unrestricted-domain system, where all parameters referred to phrase provided worse results.

We have made experiments using the thermometer-type codification instead of the floating point but, again, all results were slightly worse.

The summary of most relevant results is shown in Table 1. All of them correspond to the best network, which used 10 neurons. This is another difference with the unrestricted-domain system: we can use more neurons without overtrain symptoms. The results for the train and test columns are expressed as relative RMS. "I" means improvement, and is the average for the test set for three different number of neurons.

Experiment	Train	Test	I
Reference experiment	0.5558	0.5580	
1- Ref. + window of 5 stress	0.5581	0.5555	0.96
2- Ref. + window of 5 phonemes	0.5341	0.5355	4.65
3- Ref. + position in relation to stress	0.5463	0.5450	2.60
4- Ref. + diphthong	0.5504	0.5515	1.55
5- Ref. + syllable begins with vowel	0.5487	0.5462	2.47
6- Ref. + function word	0.5431	0.5451	2.53
7- Ref. + position of P in S	0.5537	0.5523	1.03
8- Ref. + position of S in W	0.5456	0.5462	2.29
9- Ref. + position of W in PHR	0.5440	0.5427	2.49
10- Ref. + number of P in S	0.5512	0.5494	2.21
11- Ref. + number of S in W	0.5497	0.5501	2.14
12- Ref. + number of W in P	0.5403	0.5403	3.43

Table 1. Summary of results (average for all sentences) (P=phoneme, S=syllable, W=word, PHR=phrase).

We have been able to find the right codification for all the parameters, as there is a improvement for all of them. In contrast with the results obtained in the unrestricted-domain [1], the differences for most of them are significant.

3.5 Putting everything together

In Table 2 we can see the summary of results using the parameters together. Numbers refer to items in Table 1.

Experiment	Train	Test	I
13- Ref. + 2 + 3	0.5167	0.5214	6.94
14- 13 + 4 + 5 + 6	0.5131	0.5215	6.43
15- 14 + 7 + 8 + 9	0.5027	0.5121	8.28
16- 15 + 1	0.5020	0.5062	8.51
17- 16 + 10 + 11 +12	0.4920	0.4927	11.12

Table 2. Putting everything together.

The results are really good, and, unlike the unrestricted-domain system, this system keeps improving for both the train and the test set as we increase the number of parameters, which shows the correct learning of the networks.

3.6 Increasing the size of the networks

In the unrestricted-domain system, there were overtrain symptoms with very few neurons. In our system we observed that with 10 neurons there was no overtrain. So, we decided to increase the number of neurons for our best system (experiment 17 from Table 2). We experimented with 14, 17, 20, and 24 neurons (we uses two hidden layers too, but again the results were worse). This is our best result:

Experiment	Train	Test	I
18- 17 with 20 neurons	0.4481	**0.4536**	18.71

In our best experiment the average absolute error is 390 samples, equivalent to **12.2 ms**, which is a really good figure, and, as we expected, is better than the 14.3 ms we obtained in the unrestricted-domain system.

We applied the T-Student test to the comparison of the normalized duration obtained with all the systems, and most of them are significantly different, specially when we compare experiments 12-18 with the reference one.

3.7 Comparison to rule-based system

Using a multiplicative model for duration, with the best parameter coding of the ANN experiments, the absolute error was 19.8 ms, which is clearly worse than the result obtained with our neural network.

4. CONCLUSIONS

With our working environment and the experience obtained in [1], we have been able to develop a duration model for a new female voice in a very short time. So, we demonstrated that it can be easily adapted to specific contexts and/or new databases.

The results obtained in the restricted-prosody domain show improvements that are much more significant than in [1], just as we could expect because the database is more homogeneous. The best relative RMS in [1] was 0.76428, which is clearly worse than the best result here (0.4536). This metric can be used to compare different databases.

Another important aspect is that the results improve when we include all the parameters and increase the number of neurons, a tendency we did not observe in the unrestricted-domain system.

5. REFERENCES

[1] Córdoba, R., J.A. Vallejo, J.M. Montero, J. Gutiérrez-Arriola, M.A. López, J.M. Pardo, "Automatic modeling of duration in a Spanish text-to-speech system using neural networks". Eurospeech 99, vol. IV, pp. 1619-1622.

[2] Montero, J.M., R. Córdoba, J.A. Vallejo, J. Gutiérrez-Arriola, E. Enríquez, J.M. Pardo. "Restricted-Domain Female-Voice Synthesis in Spanish: from Database Design to ANN Prosodic Modeling". ICSLP 2000.

[3] De Tournemire, S., "Identification and automatic generation of prosodic contours for a text-to-speech synthesis system in French", Eurospeech 97, pp. 191-194.

[4] Quazza, S., Heuvel, H.. (1997). The use of lexica in text-to-speech systems. *Course at Nijmegen University.*

[5] van Santen, J.P.H., "Prosodic modeling in text-to-speech synthesis", Eurospeech 97, KN – 19-27.

[6] van Santen, J.P.H., "Assignment of segmental duration in text-to-speech synthesis", Computer Speech and Language (1994)8, pp. 95-128.

[7] Boëffard, O., F. Emerard, "Application-dependent prosodic models for text-to-speech synthesis and automatic design of learning database corpus using genetic algorithm". Eurospeech 97, pp. 2507-2510.