Intelligent Automation and Soft Computing, Vol. 15, No. 4, pp. 631-646, 2009

NOVEL APPLICATIONS OF NEURAL NETWORKS IN SPEECH TECHNOLOGY SYSTEMS: SEARCH SPACE REDUCTION AND PROSODIC MODELING

J. MACIAS-GUARASA[1], J.M. MONTERO[2], J. FERREIROS[2], R. CORDOBA[2], R. SAN-SEGUNDO[2], J. GUTIERREZ-ARRIOLA[3], L.F. D'HARO[2], F. FERNANDEZ[2], R. BARRA[2] AND J.M. PARDO[2]

[1]*Department of Electronics*
University of Alcala. Spain

[2]*Speech Technology Group*
Department of Electronic Engineering
ETSIT de Telecomunicación
Universidad Politécnica de Madrid. Spain

[3]*Department of Circuit and Systems*
EUIT de Telecomunicación
Universidad Politécnica de Madrid

ABSTRACT—Neural networks (NNs) have been extensively used in speech technology systems. In this paper, we present two novel applications of NNs in speech recognition and text-to-speech systems.

In very large vocabulary speech recognition systems using the hypothesis-verification paradigm, the verification stage is usually the most time consuming. State of the art systems combine fixed size hypothesized search spaces with advanced pruning techniques. We propose a novel strategy to dynamically calculate the hypothesized search space, using neural networks as the estimation module and designing the input feature set with a careful greedy-based selection approach. The main achievement has been a statistically significant relative decrease in error rate of 33.53%, while getting a relative decrease in average computational demands of up to 19.40%.

The prosodic modeling is one of the most important tasks for developing a new text-to-speech synthesizer, especially in a female-voice high-quality restricted-domain system. Our double objective is to get accurate predictors for both the fundamental frequency (F0) curve and phoneme duration by minimizing the model estimation error in a Spanish text-to-speech system, by means of a neural network estimator, which has proved to be an excellent tool for the modeling. The resulting system predicts prosody with very good results (for duration: 15.5 ms in RMS and a correlation factor of 0.8975; for F0: 19.80 Hz in RMS and a relative RMS error of 0.43) that clearly improves our previous rule-based system.

Key Words: Speech recognition, neural networks, search space reduction, hypothesis-verification systems, greedy methods, feature set selection, prosody, F0 modeling, duration modeling, text-to-speech, parameter coding

1. INTRODUCTION

Neural networks have been extensively used in speech technology systems, both in automatic speech recognition [1][2], and text to speech conversion [3][4], with results comparable to traditional techniques, usually at a lower cost and making full use of the intrinsic discrimination capabilities of NNs. In this paper, we present two novel applications of NNs in speech recognition and text-to-speech systems.

1.1 Search Space Reduction in Automatic Speech Recognition Systems

Computational demands are one of the main factors to take into account when designing systems supposed to operate in real-time, especially when talking about public information services using the telephone network. Telephone information service providers are demanding systems and algorithms that allow them to increase the number of active recognizers to run in dedicated hardware, to be able to significantly decrease production costs.

According to this scenario, state of the art systems are usually based in some form of *progressive search* [5], whereby successively more detailed (and computationally expensive) knowledge sources are brought to bear on the recognition search as the hypothesis space is narrowed down. This approach is a generalization of the hypothesis-verification paradigm, with several cascaded stages. In the simplest case the first stage (hypothesis), a *rough analysis* module with low computational demands, face the whole search space of the task, and select a subset of this search space to be fed to the second stage (*detailed analysis* module, verification), much more demanding in computational resources and more able to accurately decode the input speech. For the whole system to be successful, the rough analysis module must ensure that the selected subset of the search space contains the right hypothesis with high probability, so as not to degrade the overall performance.

In hypothesis-verification systems, the main concern is reducing the hypothesized search space as much as possible, and this is not an easy task, especially when low detailed acoustic models are used in the preselection stage. Traditionally, these systems use a fixed size hypothesized search space, estimated according to the results obtained during system development so that a minimum error rate is achieved. Under these constraints, most of the research work aimed at lowering computational requirements has been centered in search space pruning techniques, usually based in beam search techniques [6].

The first goal of this study is focused in the hypothesis stage: instead of only relying in static or dynamic pruning techniques in the verification module, we want to design a procedure so that the hypothesized search space varies in size, different for every utterance, depending on any know-in-advance system parameter. If we lower the *average* hypothesized search space size while keeping the error rate performance, the computational demands of the overall system would be lower. Thus, the key factor to evaluate the effectiveness of different methods is calculating the reduction in average hypothesized search space size (which we will refer to as *average effort*) while keeping the required error rate. In addition to that, if the neural network estimation is accurate enough, we could even get improvements in the system error rate, and this actually happens in the experiments described below.

1.2 Prosodic Modeling in Restricted-Domain Text-to-Speech Systems

The second goal of this study was to develop an automatic system to model prosody for a Spanish text-to-speech system (TTS) in a restricted-domain environment for a female voice. This work is the continuation of [4] and [15] which were dedicated to a general-domain database for a male voice and [16], that included the first version of the restricted-domain modeling, achieving better results than our original rule-based system. For modeling duration this rule-based approach

follows a classic multiplicative Klatt model; for the F0 curve (the temporal evolution of the vocal folds vibration frequency), it is modeled a parametric way as a series of text-dependent F0 peaks and F0 valleys [17].

Although a domain-specific application does not require as many sentence structures as a general one (the delivered messages are syntactically constraint), there can be many words embedded in them (e.g., more than 40,000 family names, more than 30,000 village names, etc.). 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 if we mimic the natural prosody exhibited by the speaker.

Many studies have been successfully carried out lately in the field of automatic estimation of the prosodic values, using different techniques and input parameters to obtain the model. For duration, these automatic techniques are mainly of two types: decision trees and neural networks (the objective of this paper); another line of investigation with very good results is the statistical sum-of-products method. For F0 modeling, the dominant techniques are artificial neural networks and k-nearest-neighbor, combined with a parametric model of the F0 curve [18].

In all the systems, regardless of the modeling technique, it is crucial to find the parameters (or features) that are most significant for prosodic modeling. So, we can take advantage of previous studies dedicated to prosody prediction, but using other techniques to decide the parameter set. Neural networks have previously been used with success. In [19] a neural network was trained to predict syllable timing. In [20] they compare the performance of neural networks and Classification and Regression Trees (CART) techniques for six different languages, including Spanish. The results for both are very similar, which shows that any of them can be used. Regarding the application of these techniques to Spanish, there are very little references and none is dedicated to neural networks or CART approaches. We have considered some of them but only to decide the parameters to be used as input. See in [4] a summary of references for Spanish.

1.3 Organization of the Paper

The paper is divided into two main sections; describing the two applications of neural networks we propose (Sections 2 and 3). In both cases, there is a general structure describing the system in which the NN-based estimator is being used, the experimental setup, the methodology applied in the neural network system development, the feature selection process and the experimental results. Finally, Section 4 draws the main conclusions, Section 5 the acknowledgements and section 6 includes the references used along the paper.

2. EFFICIENT NN-BASED SEARCH SPACE REDUCTION IN A LARGE VOCABULARY SPEECH RECOGNITION SYSTEM

2.1 System Overview

The general architecture we are working on is shown in Figure 1, in which an estimator module is in charge of deciding the size of the hypothesized search space, to be passed to the detailed analysis module, using for that purpose a certain set of features extracted from the feature extraction and rough analysis processes. This estimator module will be the one in which we will use NNs as the estimation strategy. The main hypothesis generator modules are fully described in

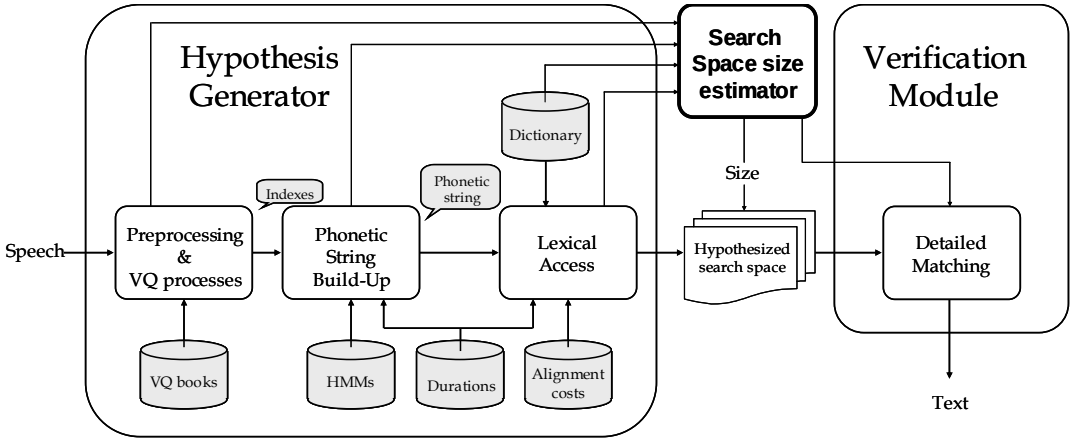

Figure 1. Speech recognition system architecture

[7], and to summarize, the current implementation of the hypothesis module follows a bottom-up, two stage strategy (a phonetic string is first generated and then matched against a tree-structured dictionary, using dynamic programming algorithms [8] and [9]).

In general, and given the proposed task, it is clear that the computational requirements are closely related to two different factors: modeling complexity and search effort in the hypothesis and verification algorithms. In this paper, we will focus our description to the work in the preselection (hypothesis generation) system, although additional reductions in computational demands are achieved in the verification module through the use of beam pruning techniques. Our target objective will be achieving a maximum error rate, which should be lower than 2% in all cases, so as not to limit the final recognition accuracy achieved by the verification stage.

2.2 Experimental Setup and Baseline System

Experiments have been carried out using part of VESTEL, a realistic isolated word telephone speech database [10], captured using the Spanish Public Switched Telephone Network and composed of 9,720 utterances. We have used a ten-fold cross-validation strategy in order to increase the statistical significance of the results and with 3 non-overlapping sets (80% of the data is used for training, 10% for validation (over-fitting detection) and tuning, and the remaining 10% for the evaluation). The validation subset is used for early stopping and to make decisions regarding the best feature selections, the optimal number of iterations, etc. All the training-validation-testing procedure is repeated for each of the 10-fold subgroups and the results are finally averaged.

The real-world application task was designed for the research and development division of the Spanish Telephone Company (Telefónica I+D) around the *white pages* idea, so that the dictionary used in this work is composed of 10,000 words, the most probable names and surnames in Spanish. The experiments will be carried out in the context of a large vocabulary isolated word recognition system. In this case, the hypothesis module will generate a *preselection list* composed of the most probable words (candidates) given the input speech utterance. The preselection list length (*PLL* from now on) used, on the average, would give us the *average effort* for the task. To give an example, the full search space would imply a preselection list composed of all the words in the dictionary (10000 words), and our objective is reducing this number of words in the preselection list (let's say down to 900 words, leading to an overall reduction of 91% in search space size), the ones that would be finally forwarded to the verification stage.

The baseline experiment uses fixed *PLL*s, which is equivalent to a fixed search space size. For its evaluation, we calculated the *inclusion error rate* achieved for every possible length of the preselection list. The inclusion error rate is obtained assuming a recognized word is within the first N candidates (N equals the PLL) proposed by the hypothesis module. In general, and given the unequal performance of recognition systems depending on the word to be recognized, that fixed length must be assigned a large value, leading to a large average effort, a large average wasted effort and to computational requirements higher than desired. Actual system performance measurements showed that the wasted effort almost equals the required average effort, so that great improvements could be achieved.

In our system, we obtained 2% error rate for a fixed PLL of around 10% of dictionary size, which is 1000 candidates. Taking into account this result and previous experiments, we established the baseline system as the one that used exactly 1000 candidates for the fixed PLL, which lead to an inclusion error rate of 1.72%.

So, our target will be achieving at least the same performance (1.72% error rate) while, reducing the average *PLL* (which equals to the average hypothesized search space size, thus lowering the computational demands for the whole system), as we will use variable *PLL*s estimated using a neural network (NN).

2.3 Neural Network System Development Methodology

2.3.1 Topology of the Neural Network

In all our experiments we will use a multi layer perceptron [12], with a single hidden layer and sigmoids as the activation functions. In order to increase the generalization capabilities of the NN, we kept the number of neurons as low as possible, leading to relatively simple topologies with less than 600 weights to train.

2.3.2 Feature Inventory and Input Coding

In our case we have created a wide spectrum of possibilities regarding the available feature set: We designed an inventory of 56 features that can be classified in three broad classes:

- Direct parameters: Obtained from the characteristics of the acoustic utterance or the preselection process: number of frames, phonetic string length, acoustic search score, lexical access costs, etc.
- Derived parameters: Calculated from the previous ones applying different types of normalization schemes (dividing by number of frames, phonetic string length, etc.)
- Lexical Access Statistical Parameters: Averages and standard deviations calculated over the lexical access costs distribution, for different PLLs.

The input coding schemes we tested include both single and multiple inputs per parameter:

- For parameters coded in a single input, the alternatives were: No coding (raw data input), linearly scaling the full parameter range between a minimum and maximum value of the input neuron, standard z-score normalization, with optional data clipping to some predefined values, proportional to the standard deviation of the training data set.
- For parameters coded in multiple inputs:
  1. Using a uniformly distributed linear mapping function: dividing the full parameter range by the number of inputs, so that we activate the input corresponding to the range in which the parameter value is located.

2. Using a non-uniform mapping function: considering the distribution of the parameter values in the training database, the segments assigned to each input are chosen so that the number of activations is equalized for each parameter.

In addition to that, we tested different number of input units per parameter and different values to encode every input activation. We tested all the different coding schemes using a subset of the training database and we established that standard *z-score* normalization for the input features achieved the best results [11].

2.3.3 Output Coding and NN Post-Processing

Our network is aimed at estimating a certain *PLL*, given the input parameters. For coding the activations of the output neurons we could use the same strategies we discussed above for input parameter coding. In this case we are interested in multiple output neurons, as they could encode *PLL* values in a better way. In output coding, however, using a uniformly distributed linear mapping function lead to very bad results, as only the first few neurons are activated during training, as most utterances are recognized for the first few candidates in the preselection list.

We evaluated all the possible output coding strategies and we established that the best results where obtained with the following setup [11]:

- Every output neuron k is defined to represent a different PLL range (PLLs from *lowerSegmentLength(k)* to *upperSegmentLength(k)*), leading to the task formulated as a classification problem in which the NN should decide which is the most likely output neuron to be activated.
- The *PLL* ranges that every output neuron represents are trained with a criterion that aims to get, when possible, a uniform number of training samples for all of them, in order to avoid data sparseness during training.

The NN output values are finally post-processed to obtain the final *PLL*. The idea is further increasing the proposed length, so that mismatches between the training and testing sets are compensated to a certain extent. Different alternatives where tested:

- The output neuron with the higher activation value decides the *PLL* to be used (the upper limit of the *PLL* range associated to the winning neuron). If *act(k)* is the activation value for the *k* output neuron:

$$PLL = upperSegmentLength(g), \quad g = \underset{0<k\leq numOutputNeurons}{\arg\max} \left[act(k)\right] \tag{1}$$

- The *PLL* is calculated as a linear combination of output neuron activations multiplied by the upper limit of the *PLL* range associated to each output neuron. The rationale for this approach is based on the fact that, given certain premises, NN outputs can be interpreted as class posterior probabilities [12], so that all output neurons have something to say regarding the estimated *PLL*:

$$PLL = \sum_{k=1}^{NumOutputNeurons} upperSegmentLength(k) \cdot act(k) \tag{2}$$

From these basic approaches, two additional mechanisms were tested to increase system robustness: Adding a fixed threshold to the proposed *PLL*. If *PLL** is the final *PLL* to be used:

$$PLL^* = PLL + fixedTrainedThreshold \tag{3}$$

or using a proportional threshold to the proposed PLL:

$$PLL^* = PLL \cdot (1 + proportionalTrainedThreshold) \quad (4)$$

Of course those thresholds are also calculated during the training phase, imposing the achievement of a certain inclusion rate. We tested all the NN post processing strategies, and the best one proved to be the one given by equation (2), with the threshold equation (3) [13].

2.4 Feature Selection

The initial experimentation described in sections 2.3.2 and 2.3.3 gave us the experimental scenario to be used in the feature selection process. In order to select the most discriminative features for our task, we used an adapted version of the *greedy* algorithm [14]. Initially, this procedure was planned as follows:

1. The feature set is initialized as empty.
2. In every iteration of the feature selection algorithm, feature ensembles are generated adding every pending feature to the existing feature set.
3. Experiments with variable number of iterations are performed for every feature ensemble.
4. The feature ensemble achieving the highest reduction in preselection error rate for the optimal number of iterations is selected as the new feature set for the next iteration.
5. Continue with step 2 if the preselection error rate decreases.

Initial evaluation showed that the computational complexity of step 3 is huge, so that we simplified the process as follows:

- In steps 3 and 4 we select the 8 feature ensembles which lead to the best results using a training procedure with the optimal number of iterations found *in the previous iteration*.
- Before step 5 we carry out experiments to determine the optimal number of iterations for each of the 8 best ensembles and we finally select the ensemble with the highest reduction in preselection error rate.

With this approach, a set of 4 features was selected as the optimum. The most discriminative features are related to the standard deviation of the lexical access costs, the normalized acoustic score from the phonetic string build up module and the phonetic string length.

2.5 Experimental Results

As discussed above, the NN-based system will generate a different *PLL* for every utterance. Inclusion error rates are calculated according to this approach and can be directly compared to the baseline system inclusion error rate (1.72%). On the other hand, computational requirements in the NN-based system are measured computing the *average* estimated *PLL*, and will be compared with the fixed size of 1000 candidates in the baseline system.

Obviously we still need relative quality measurements, so that we will also calculate relative error rate and average effort increments (Δ) when using the NN-based system. Table I shows the final results (along with 95% confidence intervals for the error rate figures).

Table I. Experimental results

	Inclusion error rate	**Average effort**	**Δ error (% relative)**	**Δ effort (% relative)**
Fixed list length system	1.72% ± 0.21%	1000	-	-
NN based system	1.14% ± 0.26%	806	-33.53%	-19.40%

So, the best NN based system achieves a 33.53% reduction in error rate, while also reducing the average computational effort in almost 20%. In addition to that, the differences are statistically significant (confidence intervals do not overlap).

The computational impact of the NN calculations is negligible when compared to the overall runtime of the preselection stage (under 0.01% of the total runtime).

3. PARAMETER SELECTION FOR PROSODIC MODELLING IN A RESTRICTED-DOMAIN SPANISH TEXT-TO-SPEECH SYSTEM

3.1 System Overview

The general architecture we are working on is shown in Figure 1, in which an estimator module is in charge of deciding the prosody to apply in the generated speech signal. The inputs to the prosody estimation module are generated by the natural language processing stages, and are typically related to the phonetic context of the input text, the syllabic structure, the relative position of the phonetic or lexical unit being considered, etc.

3.2 Experimental Setup

The database used in this paper is described in [16]. We extracted a set of 19 Carrier Sentences (CS) from two real services in banking and traffic information domains, provided by the IVR design company. The CS contained 24 Variable Fields (VF) and each VF conveys the most important information in the CS and must be surrounded by compulsory pauses. Prosodic values are only computed for the VFs. We classified the CS into 3 classes or groups:

- Proper Names: surnames (both compound and simple ones), cities, villages, etc
- Questions: bank-related information such as currency, check status, etc.
- Noun Phrases: regarding accounts, credit cards, names of transactions and banks...

For the design of the database we used a greedy algorithm that is described in [16]. We aimed at selecting a small database with the same probability distributions of certain phonetic and prosodic features as in a very big database (about 6600 phonemes and 2800 syllables per class)

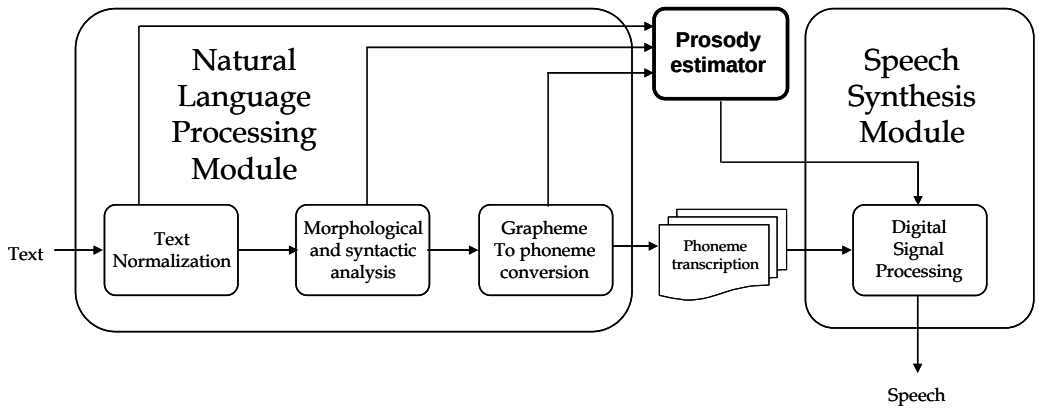

Figure 2. Text-to-speech system architecture

3.3 Neural Network System Development Methodology

3.3.1 Topology of the Neural Network

For both duration and F0 modeling, we have used a multilayer perceptron (MLP), using the sigmoid as the activation function and the backpropagation algorithm for training. For each phoneme (or syllable), we compute a series of parameters (features), which we code and use their values as inputs to the neural network. There is one output in our networks: the duration of the phoneme (or the F0 of the syllable). For duration experiments we used 2 sets (training and testing) and we divided the training into three phases of 300 iterations each (for over-fitting detection). For F0, we used a ten-fold cross-validation strategy with 3 non-overlapping sets (one for training, one for over-fitting detection and one for the final evaluation). As it is very difficult to know the optimum number of neurons and layers that the net should have, a set of experiments were carried out in order to optimize the system without overtraining.

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 [4] we compared both approaches. The 3-networks option improved the results in 6% for duration so we decided to use 3 different networks in our duration experiments.

3.3.2 Feature Input Coding

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

1. Binary coding: this is the standard coding for true/false parameters.
2. One-of-N coding: to code N classes, we use N neurons and only 1 of them is active.
3. In ordinal values we have more possibilities, as these values can be ordered:
4. Percentage transformation: we divide the current value by the maximum value to obtain a percentage. We obtain a floating-point value between 0 and 1 as input.
5. Thermometer: we divide all the possible values into different classes (intervals). We activate all the neurons until we get to the current class and leave the remaining neurons inactive. We developed an algorithm to obtain a uniform distribution of all the classes.
6. Z-Score mapping: we normalize the floating-point value by accounting for the average and the standard deviation of the variable (a good coding for very variable parameters).

3.3.3 Output Coding

We obtained in [4] that phoneme durations should be normalized by the duration of the phrase (to be less affected by changes of speed in the database recordings). After the normalization, we use the standard deviation of the logarithm of the duration (to balance the distribution of the values and to minimize the error, as it includes the characteristic duration of each phoneme in the prediction) and a Z-Score codification. For F0, we just used Z-Score.

3.3.4 Network Evaluation

To evaluate the error of the networks (difference between the prediction from the network and the optimum value), we have considered different metrics. The most important one is the Root Mean Square error (RMSE). Another one is the relative *RMSE* (RMSE / sqrt [Σ [t-t_i]^2]), that it is adimensional and independent of the way we code the target values (t_i), and it does not have an offset.

3.4 Feature Selection and Experimental Results

3.4.1 Base Experiment for Duration

In our base experiment for duration (first row of Table I) we have decided to include just the phoneme identity (with a set of 38 phonemes and a windowing of three values, described in next section), and the stress, which are the most relevant parameters according to our previous work and to our own statistical studies. The coding used is a one-of-n coding: a '1' in the input which corresponds to the phoneme and '0' for all the other inputs.

In Table I we can see the relative RMSE and the average improvement obtained for the test set with individual parameters, using a 10-neurons network. The last column shows the results of applying a T-Student test to compare the base experiment and the experiment considered (when "2-tail-sig" is below 0.05 the difference between both systems is statistically significant).

3.4.2 Contextual Phonemes

In our previous studies, the duration of a phoneme was significantly affected by the phoneme to the right and to the left. As the number of phonemes is too high, we made 14 clusters of phonemes according to its type. Using a two-phonemes context (a window of five values) we obtained an improvement of 5% for the test set (Table I, experiment 1). This result is really remarkable, as it shows the importance of contextual information. But for a 7-values window the results were slightly worse.

3.4.3 Parameters Related to Position and Binary Parameters

In [4] we found that "Position in phrase in relation to first/last stress" was an especially relevant parameter, as it explicitly includes the "lengthening before pause" effect. We coded each syllable in 5 possible classes with very good results (Table I, experiment 3).

We have also obtained new significant improvements over the base experiment by considering several binary parameters (experiments 4-6 in Table I):

- Syllable structure: syllables ending with a vowel (open syllables) are generally longer.
- Vowel in diphthong ("i/u" before/after "a/e/o"). In Spanish, we differentiate both of them as different allophones, and they follow different rules for duration.
- Phoneme in a function word. Syllables in a function word are shorter.

In [4] we considered different alternatives for parameters related to position and decided to use: phoneme in the syllable, syllable in the word, and word in the phrase, as they provide different information to the network (not redundant), their range of values is smaller, and, so, fewer neurons and classes are needed. We carry out the following steps for the coding:

1. To normalize the value of position by the total length of the higher-order unit
2. This value is coded using 3 classes, and their intervals are computed automatically.
3. The 3 classes use a thermometer-type coding with 2 inputs (number of classes minus 1).

The results of these experiments (7 to 9 in Table I) have improved the base experiment again. The best parameter 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 were worse. The reason is that the range of values is much more uniform in this restricted-domain system.

3.4.4 Parameters Related to the "Number of Units"

In a similar way as for parameters related to position, we decided to use the number of phonemes in the syllable, the number of syllables in the word, and the number of words in the phrase. Because of their different distribution, we needed a different coding:

1. To normalize the value by the maximum one: a floating point value between 0 and 1.

2. To apply Z-score (using average and standard deviation): this way, we can restrict at our will the operating range of the parameter, which is too variable.

The improvements (experiments 10-12 of Table I) were significant and very similar to those of position parameters (the number of words in the phrase is the best parameter). In order to check the suitability of this floating point coding, we tested the thermometer-type coding (as for position-related parameters), but the results were always below.

3.4.5 Summary of Results for Duration

The summary in Table II corresponds to the best network (10 neurons). We have obtained the best results for: window of 5 phonemes, number of words in the phrase, position of the word in the phrase and position in phrase in relation to first/last stress.(Stress is important too, but it is included in the base experiment); almost all the improvements are significant (not as in [4]).

Table II. Summary of results in average relative RMS (for duration)

Experiment	Test set	Improvement	2-tail-sig
Base experiment	0.5580	-	-
1- Base + window of 5 phonemes	0.5318	4.98 %	0.000
2- Base + window of 7 phonemes	0.5350	4.81 %	0.000
3- Base + position in phrase	0.5450	2.48 %	0.001
4- Base + vowel in diphthong	0.5515	1.53 %	0.045
5- Base + syllable structure	0.5462	2.43 %	0.001
6- Base + function word	0.5451	2.35 %	0.000
7- Base + position of Phoneme in Sentence	0.5523	1.03 %	0.419
8- Base + position of Sentence in Word	0.5462	2.29 %	0.006
9- Base + position of Word in Phrase	0.5427	2.49 %	0.001
10- Base + number of Phoneme in Sentence	0.5494	2.07 %	0.010
11- Base + number of Sentence in Word	0.5501	2.20 %	0.048
12- Base + number of Word in Phrase	0.5403	3.43 %	0.000

3.4.6 Final Experiments for Duration

The next set of experiments was dedicated to including all the parameters together. This is the crucial step in neural networks, because many times the improvements combining parameters are not additive, because the parameters are closely correlated (do not offer additional information), or the topology of the network needs to be tuned (a larger number of neurons may be needed).

In Table III we can see the summary of results. The numbers in the description of the experiments refer to the experiments specified in Table II. The T-Student test is now applied to the comparison of an experiment with the previous one.

Table III. Results for duration including all parameters.

Experiment	Test	Improvement	2-tail-sig
Base experiment	0.5580	-	-
13- Base + 1 + 3	0.5214	**6.58 %**	0.000
14- 13 + 4 + 5 + 6	0.5206	6.83 %	0.512
15- 14 + 7 + 8 + 9	0.5121	**8.09 %**	0.039
16- 15 + 10 + 11 + 12	0.4927	**11.12 %**	0.002

- Experiment 13: it is the base experiment using now a window of 5 phonemes and position in phrase in relation to first/last stress. The improvement was remarkable.
- Experiment 14: we added the binary parameters: vowel in diphthong, syllable structure and function word. The improvement is reduced and not significant
- Experiment 15: with position parameters. The improvement is significant.
- Experiment 16: including the 'no. of units' parameters with significant improvements.

The results are really good, and the 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.

In the unrestricted-domain system [4], there were symptoms of overtraining with very few neurons, which impeded the improvement of the global system. In this system, the best results correspond to the topology with 20 neurons. The improvement over the base experiment is 18.71%, which shows that our solutions improved this system drastically. The relative RMS is 0.4536, the average absolute error is 11.79 ms, and the absolute RMS is 15.5 ms. The Pearson correlation coefficient between estimated and measured durations is 0.8975, a very good figure.

3.4.7 Comparison to Previous Systems

As could be anticipated, the results are much better than those obtained with the unrestricted-domain database: an absolute RMS equal to 19.1 ms. The relative RMS was equal to 0.76428, clearly worse than the 0.4536 obtained in this domain.

Using our previous multiplicative rule-based system, with the best parameter coding of the ANN experiments, the absolute error was 19.8 ms and the RMS was 28.5 ms, which is clearly worse than the result obtained with our neural network.

Regarding other works in the literature, in [4] a comparison with several systems is included, but as explained there, the comparison is not fair in any case as the corpus used in the papers are completely different.

3.4.8 F0 Experiments

For F0, we performed similar experiments with a different set of parameters. Our previous rule-based system used features such as: whether the syllable is stressed, whether the following syllable is stressed, the type of punctuation mark at the end of the intonation group (this parameter is related to the shape of the F0 curve at the end of the group) and the number of stressed syllables and the position of the syllable in the group. The F0-curve obtained this way is acceptable but unnatural in human perception tests [15].

In addition to these general parameters, we tried several ways of coding the influence of the carrier sentences from the restricted-domain. The best results obtained correspond to a one-of-N coding of the carrier sentences (we grouped sentences according to 3 classes as defined in section 3.1; with only a 1% improvement, that is not significant). No significant improvement was obtained through parameters related to position, to function words or to the number of units.

The summary in Table IV corresponds to the best network (20 neurons). We have obtained the best results for: a one-of-N coding for the carrier sentence and the final punctuation mark, a window of 11 syllables for stress and for the position of the syllable in the phrase (in relation to first and last stressed syllable). All the improvements are significant when compared to the previous one except for experiments 5 and 6.

4. CONCLUSIONS AND FUTURE WORK

The main conclusion of the presented paper is that the use of neural networks has proved to be an excellent alternative to traditional methods in novel applications related to speech technology systems:

Table IV. Results in average relative RMSE

F0 Experiment	Test	Improvement
Base experiment: stress	0.7378	-
1- stress in a 3-syllables window	0.6815	7.63 %
2- stress in a 11-syllables window	0.6326	14.26 %
3- 2 +final punctuation mark	0.5500	25.45 %
4- 3 + identifier of the carrier sentence	0.4554	38.28 %
5- 4 + position of the syllable in the group	0.4360	40.91 %
6- 5 + 3-neural-networks option	0.4312	41.56 %

- In the case of estimating the hypothesis search space size, the proposed NN-based system clearly outperforms the baseline system (33% reduction in error rate and 20% reduction in computational demands) and with statistically significant results. We also presented a carefully designed experimental methodology, using a greedy-based strategy for feature selection and an optimal experimental setup regarding input and output coding, along with a NN output post-processing system that relies in the interpretation of the NN outputs as being class posterior probabilities. Given the good results obtained in our classification networks, we have started a study on the estimation of word confidence measures, to allow assessing the reliability of the recognition systems used. Initial results are really encouraging, as we are outperforming traditional methods using parameters related to acoustic scores with some of the ones proposed in this paper.
- In the case of the prosody estimation task, when comparing to our previous rule-based systems, the results are much better, even when using a limited number of parameters. As we expected, the results obtained in the restricted-prosody domain show improvements that are much more significant than in [4] (because the database is more homogeneous) and than in [16] (due to a better parameter selection): for duration: 15.5 ms in RMS and a correlation factor of 0.8975; for F0: 19.80 Hz in RMS and a relative RMS error of 0.43. For a new female voice, we have demonstrated that our prosodic model can be easily adapted to specific contexts and/or new databases in a very short time. For duration 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. Regarding the topology, it is difficult to find the optimum of the network. It is better to begin with a low number of neurons and increase it step by step. The same applies to the inclusion of parameters: it is better to decide their best coding in small networks. We have found that a second hidden layer is not necessary. The "Z-score" normalization for numeric parameters shows a good behavior: it adjusts the margin of accepted values automatically rejecting the out-of-range values. In general, we can say that we have found a good compromise between network topology and parameters considered, with good results that are stable. The system has been included in a commercial high quality TTS system in Spanish [16] [21].

ACKNOWLEDGEMENTS

This work has been partially supported by the following projects: EDECAN (MEC ref: TIN2005-08660-C04), ROBINT (MEC ref: DPI2004-07908-C02), TINA (UPM and DGUI-CAM ref: R05/10922), and ATINA (UPM and DGUI-CAM ref: CCG06-UPM/COM-516). The work presented here was carried out while Javier Macias-Guarasa was a member of the Speech

Technology Group (Department of Electronic Engineering, ETSIT de Telecomunicación. Universidad Politécnica de Madrid).

REFERENCES

1. H. Bourlard y N. Morgan. “Connectionist speech recognition—A hybrid approach”. Kluwer Academic, 1994.
2. A. J. Robinson, G. D. Cook, D. P. W. Ellis, E. Fosler-Lussier, S. J. Renals, and D. A. G. Williams. “Connectionist speech recognition of broadcast news”. Speech Communication, 37:27-45, 2002.
3. J. Burniston and K.M. Curtis. “A Hybrid Neural Network/Rule Based Architecture for Diphone Speech Synthesis”. In International Symposium on Speech, Image Processing and Neural Networks Proceedings. 323-6, 1994.
4. R. Córdoba, J.M. Montero, J. Gutiérrez-Arriola, J.A. Vallejo, E. Enríquez, and J.M. Pardo. Selection of the most significant parameters for duration modeling in a Spanish text-to-speech system using neural networks. Computer Speech & Language, Vol 16 Nº 2, pp. 183-203, 2002.
5. J.L. Gauvain and L. Lamel, “Large-vocabulary continuous speech recognition: advances and applications”. Proceedings of the IEEE, Volume: 88, Issue: 8, pp. 1181-1200. 2000.
6. S. Ortmanns, H. Ney, and A. Eiden, "Language-Model Look-Ahead for Large Vocabulary Speech Recognition", Proc. Int. Conf. on Spoken Language Processing, vol. 4, Philadelphia, PA, USA, pp. 2095-2098, 1996.
7. J. Macias-Guarasa, A. Gallardo, J. Ferreiros, J.M. Pardo, and L. Villarrubia, "Initial Evaluation of a Preselection Module for a Flexible Large Vocabulary Speech Recognition System in Telephone Environment". Proc. Int. Conf. on Spoken Language Processing, Philadelphia, PA, USA, pp. 1343-1346, 1996.
8. H. Ney. “The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition”. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 32, n. 2, pp. 263-271, 1984.
9. L. Fissore, P. Laface, G. Micca, and R. Pieraccini, “Lexical Access to Large Vocabularies for Speech Recognition”. IEEE Transactions on Acoustics, Speech and Signal Processing vol. 37, n. 8, pp. 1197-1213, 1989.
10. D. Tapias, A. Acero, J. Esteve, and J.C. Torrecilla, "The VESTEL Telephone Speech Database". Proc. Int. Conf. on Spoken Language Processing, Yokohama, Japan, pp. 1343-1346, 1994.
11. J. Macias-Guarasa, J. Ferreiros, J. Colás, A. Gallardo-Antolín, and J.M. Pardo, “Improved Variable Preselection List Length Estimation Using NNs In A Large Vocabulary Telephone Speech Recognition System”. Proc. Int. Conf. on Spoken Language Processing, Beijing, China, pp. 823-826, 2000.
12. C.M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press. pp. 116-161 and pp. 245-247, 1995.
13. J. Macias-Guarasa, “Architectures and Methods in Large Vocabulary Speech Recognition Systems.” PhD. Thesis. Universidad Politécnica de Madrid, 2001.
14. J. Kittler, ”Feature set search algorithms” in Pattern Recognition and Signal Processing, C.H. Chen, Ed., Sijthoff and Noordhoff, The Netherlands, pp. 41-60, 1978.
15. J.A. Vallejo. Mejora de la frecuencia fundamental en la conversión de texto a voz. PhD Thesis Universidad Politécnica de Madrid, 1998.

16. J.M. Montero, R. Córdoba, J.A. Vallejo, J. Gutiérrez-Arriola, E. Enríquez, and J.M. Pardo. Restricted-Domain Female-Voice Synthesis in Spanish: from Database Design to ANN Prosodic Modeling. Proceedings of ICSLP, pp. 621-624, 2000.
17. J. Allen, S. Hunnicut, and D.H. Klatt. From Text to Speech: The MITalk System. Cambridge University Press, Cambridge, 1987.
18. S. Tournemire. Identification and automatic generation of prosodic contours for a text-to-speech synthesis system in French. Proceedings of Eurospeech, pp. 191-194, 1997.
19. W.N. Campbell. Syllable-based segmental duration. In Bailly, G., Benoit, C., and Sawallis, T.R. (Eds.) Talking machines: theories, models and designs (pp. 211-224). Elsevier, 1992.
20. J.W. Fackrell, H. Vereecken, J.P. Martens, and B. Van Coile. Multilingual prosody modeling using cascades of regression trees and neural networks. Proceedings of Eurospeech, pp. 1835-1838, 1999.
21. Speech Technology Group online text to speech demonstration. http://www-gth.die.upm.es

ABOUT THE AUTHORS

J. Macias-Guarasa received his MSEE degree (1992) and Ph.D. (2001) degrees from Universidad Politécnica de Madrid (UPM), with highest distinctions. From 1990 to 2007 he was a member of the Speech Technology Group and associate professor at UPM. He is currently associate professor in the Department of Electronics of the University of Alcalá. He spent six months in the Speech Group of the ICSI in Berkeley, California.

J. M. Montero Martínez received his MSEE (1992) and Ph.D. (2003) degrees from Universidad Politécnica de Madrid (UPM), with highest distinctions. He spent seven months in the Speech Group of the ICSI in Berkeley, California. Currently, he is associate professor in the Department of Electronic Engineering at UPM and member of the Speech Technology Group since 1990.

J. Ferreiros López received his MSEE (1990) and Ph.D. (1996) degrees from Universidad Politécnica de Madrid (UPM) with highest distinctions. From 1988 Javier is member of the Speech Technology Group at UPM, where he holds an associate professor position and currently is the associate director of the Department of Electronic Engineering. From Oct 1999 to Apr 2000, Javier stayed at ICSI, Berkeley, CA as visiting researcher. His research interests focus on spoken dialog systems.

R. de Cordoba Herralde received his MSEE (1991) and Ph.D. (1995) degrees from Universidad Politécnica de Madrid (UPM) with highest distinctions. He is a member of the Speech Technology Group since 1990, teaching in the UPM since 1993, now working as Associate Professor in the Department of Electronic Engineering. He worked as Research Associate in Cambridge University (UK), Speech, Vision and Robotics Group, in 2001.

R. San-Segundo received his MSEE (1997) and Ph.D. (2002) degrees from Universidad Politécnica de Madrid (UPM), with highest distinctions. During 1999 and 2000, Ruben did two summer stays at The Center of Spoken Language Research (CSLR), University of Colorado (Boulder). From Sep. 2001 through Feb. 2003, Rubén worked at the Speech Technology Group of Telefónica I+D.

J. M. Gutiérrez-Arriola received her MSEE degree from the University of Cantabria in 1994. Currently, Juana is associate professor at the department of Circuit and Systems at EUITT (UPM) and member of the Speech Technology Group (GTH).

L. F. D'Haro Enríquez received his degree as Electronics Engineer in 2000, from Universidad Autónoma de Occidente in Cali, Colombia. He is currently assistant professor and Ph.D. student at UPM, Spain. In 2005 he stayed at Computer Science VI, RWTH Aachen University (Germany) working in machine translation and language modeling, and in 2006 at AT&T labs research in Florham Park, NJ (USA), working in multimodal dialogue interaction and interfaces.

F. Fernández Martínez received his MSEE degree from Universidad Politécnica de Madrid (UPM) in 2002 (with highest distinction) and he is currently assistant professor and PhD candidate at UPM. During 2006, Fernando did a summer stay at The IDIAP Research Institute affiliated with the "Ecole Polytechnique Fédérale de Lausanne" (EPFL) and the University of Geneva (Switzerland).

R. Barra Chicote received his MSEE degree from Technical University of Madrid in 2005 (with highest distinction). Since 2003 he is a member of the Speech Technology Group. In 2006 he was a visitor researcher of the Center for Spoken Language Research (CSLR) at Colorado University. In 2008 he was a visitor researcher of the Centre for Speech Technology Research (CSTR) at Edinburgh University. His main research interests are related to emotional speech synthesis and automatic emotion identification.

J. M. Pardo Muñoz (M'84-SM'04) got his MSEE Degree (1978) and PhD (1981) from Universidad Politécnica de Madrid. He got a best graduate national award (1980) and a best PhD Thesis national award (1982). He is Head of the Speech Technology Group since 1987 and Full Professor since 1992. He was head of the Electronic Engineering Dept. from 1995-2004. Prof Pardo has been a Fulbright Scholar at MIT in 1983-84, a visiting scientist at SRI International in 1986 and a visiting fellow at the ICSI in 2005-2006. He was chairman of EUROSPEECH 1995, member of the ISCA Advisory Council, ELSNET Executive Board, and NATO RSG 10 & IST 3.