VARIABLE PRESELECTION LIST LENGTH ESTIMATION USING NEURAL NETWORKS IN A TELEPHONE SPEECH HYPOTHESIS-VERIFICATION SYSTEM

*J. Macías-Guarasa, J. Ferreiros, A. Gallardo, R. San-Segundo, J.M. Pardo and L. Villarrubia**
Grupo de Tecnología del Habla. Dept. de Ingeniería Electrónica. Universidad Politécnica de Madrid. Spain
* Grupo de Tecnología del Habla. Telefónica Investigación y Desarrollo. Spain
macias@die.upm.es - http://www-gth.die.upm.es

ABSTRACT

At ICSLP'98 we presented some preliminary results on automatic preselection list length estimation using parametric and non-parametric techniques, for a flexible large and very large vocabulary, speaker independent, isolated-word hypothesis generation system in a telephone environment, with vocabularies of up to 10000 words.

In the baseline system, the preselection module generates a fixed-length list of candidate words, to be given to the verification stage. Our idea is making this length variable, depending on any known-in-advance system parameter, to allow decreasing computational demands.

In this paper we present a novel approach to preselection list length estimation. A neural network is used to give an initial estimate of the required length, which is further processed to obtain a final value. The key factor to evaluate different methods is calculating the average preselection list length (effort) while keeping the required error rate.

1 INTRODUCTION

Computational demands are one of the main factors to take into account when designing systems supposed to operate in real-time, specially 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 recognisers to run in dedicated hardware, to be able to significantly decrease production costs.

According to this scenery, systems based on the hypothesis-verification paradigm are generally used, so that the output of a rough analysis module, with low computational demands, is fed to a detailed matching module [3]. The first one generates a list of candidate words, which will be further processed with a much more detailed strategy. For the whole system to be successful, the rough analysis module must ensure that the right word is within the list it generates with high probability, so as not to degrade the overall performance.

In hypothesis-verification systems, the fine acoustic matcher is usually the most time consuming, so that the main concern is reducing the preselection list length as much as possible.

This is not an easy task, especially when low detailed acoustic models are used in the preselection stage. Traditionally, these systems use a *fixed preselection list length*, estimated according to the results obtained during system development so that a minimum recognition rate is achieved.

Using this approach, designers are obliged to use a high number of words to include in the preselection list. The idea we are considering is estimating a different preselection list length for every utterance, so that we can lower the *average effort* needed for the recognition process. For our purposes, we define it as the average preselection list length required to ensure that the error rate in this stage is under a certain level, in our case 2%. If this *average effort* were below the *fixed length* described above, while maintaining recognition-rate, the approach would effectively decrease, on an average, computational requirements (although final reduction should take into account the processing time of the detailed matching module).

In [1] the idea was using any available system parameter (number of frames, phonetic string length, phonetic string build up probability estimation, lexical access cost, etc.) to estimate the list length. Parametric and non-parametric approaches were developed and preliminary results showed promising results.

In this paper we propose a novel approach based on the use of neural networks as initial estimators, post-processing their output to increase robustness to recognition set mismatches

2 SYSTEM OVERVIEW

The main hypothesis generation modules in our architecture are Acoustic Processing, Phonetic-string build-up (PSBU) and Lexical Access (LA), using a highly modular architecture; as shown in **Figure 1**.

23 automatically clustered allophone-like context independent SCHMM are used [2], along with two models for initial and final silence.

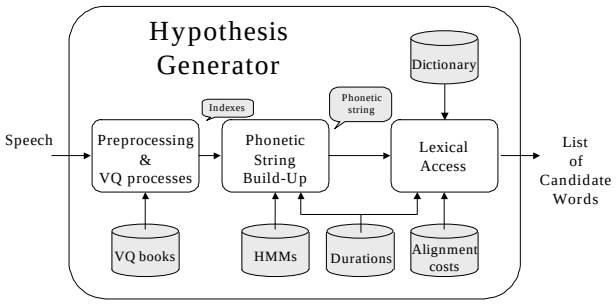

Figure 1. Preselection system architecture

3 USING NEURAL NETWORKS

The idea behind variable preselection list length estimation appeared as we observed that a certain relationship could exist between recognition accuracy and some parameters available in the recognition process.

For example, in our experiments, usually, longer words were more easily recognised than shorter ones. So, it seemed reasonable that for longer words, a shorter preselection list could be used.

1.1 Alternatives

When talking about estimation, several techniques are available with more or less success stories according to the amount of a priori knowledge available on the given task and the amount of experimental data to process.

In our case, there are no explicit algorithmic formulation of the relationship we are pursuing. Moreover, there is no theoretically supported evidence that such a relationship exists.

Traditional approaches using parametric and non-parametric techniques have been applied [1] and promising results were obtained, although an extensive test is still to be done.

In this paper, our idea was evaluating to what extent neural networks could extract the information hidden in the available system parameters, as related to estimating preselection list length.

1.2 Network topology

In our initial approach, a multilayer perceptron with a simple architecture has been tested: one hidden layer with 7 neurons, 4 input neurons and 11 classification outputs.

Some other arquitectural alternatives have been proposed, as we will discuss later.

1.2.1 Input parameters and coding

The input parameters selected initially are (from a set of up to 24 different ones): number of frames, phonetic string length, number of phones of the first candidate word and PSBU log-probability normalised by the number of frames.

All of the used parameters have been adequately coded according to different criteria. In the initial set-up simple scaling has been applied. Additional approaches to be tested include normalisation, multiple input neurons per parameters (using or not thermometer coding and binary or floating values), range clipping, etc., but they have not been applied in the experiments we are discussing.

1.2.2 Output coding

Every output is intended to indicate different list length segments. Initially, the intervals were assigned using an uniform linear distribution. For example, in the 10000 words dictionary case, every output neuron represents 10000/11=909 additional candidates to add to the preselection list. So, in the recognition stage, if the activation of neuron 3 were higher than the rest, this would mean we would need a preselection list of around 3000 candidates.

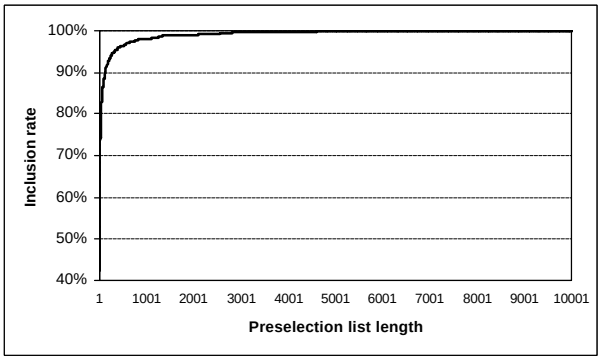

Figure 2. Inclusion rate histogram

Using this approach, and due to the shape of the preselection rate histogram (recognition rate versus number of candidates, as shown in **Figure 2**), the number of activations per output neuron available in the training set is not homogeneous.

In order to equalise the number of activations given the training samples (as far as it can be done), the *segment length distribution* (i.e. the preselection list to use in case a certain output neuron is activated) is also trained so that it's consistent with the inclusion rate histogram. In **Table 1** we show the segment limits used (the upper segment limit is actually the number of candidates used).

Table 1. Preselection list length limits used for every output

Output Neuron	Upper segment limit	Output Neuron	Upper segment limit
1	1	7	32
2	2	8	69
3	3	9	156
4	6	10	449
5	10	11	10000
6	18		

So, for example, the first output would be activated in the learning process if the word is recognised exactly in

the first position, while the last output would be activated for words recognised between positions 450 (lower limit of the segment corresponding to the last output) and 10000 (upper limit). The number of activations we get with this output coding is shown in **Figure 3**. It is clear that the situation is still not ideal, as recognition rate for the first candidate is around 50%, so that half of the available examples would activate the first output neuron!

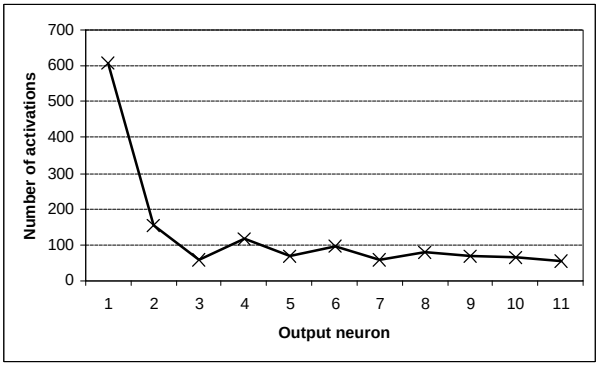

Figure 3. Number of activations per output neuron in the training set

1.2.3 Neural network training process

Standard backpropagation training using the online strategy has been applied. We have also established till what extent we were going to be able to effectively train the network parameters given the different topologies to be used and concluded that in the final experimentation process over 4000 utterances training database would be required. In this preliminary task, the 1000 utterances available are enough to ensure reasonable parameter estimation (given the network topology used).

1.3 Neural network estimation post-processing

In the recognition phase, the network output activations are post-processed to obtain the final hypothesis list length, in order to increase robustness.

The idea is further increasing the proposed length, so that mismatches between the training and testing sets are compensated to a certain extent (we are interested in achieving a given error rate so that less attention is paid to obtaining a given average effort)

The two alternatives tested when deciding the final preselection list length are:

- The *winner* output neuron decides (from now on *WN*). For example if neuron number 3 has the highest score, 3000 candidates will be used (in case we use linear homogeneous segments)
- The length is calculated as a linear combination of normalised activations multiplied by upper limit of the corresponding segment, as follows (from now on LC):

$$length = \sum_{i=1}^{NumOutputNeurons} Neuron_{length}(i) \cdot normact(i)$$

$Neuron_{length}$ is the upper limit of the corresponding output neuron (according to Table 1) and *normact* is the normalised activation of this neuron. The motivation for this formula relies in the fact that the normalised activations of the output neurons can be interpreted as posterior probability estimates, so that all neurons have something to say regarding preselection list length, and this helps to further increase robustness.

The formula can be considered to overestimate the actually required list length (according to the way the network is trained), and this is really the case. The advantage in this case, as compared to using just a fixed or proportional threshold (as discussed below) is that this overestimation is somehow more "informed" as it depends on the given network output, and we can expect it to be more consistent.

In both cases, an additional fixed or proportional threshold can be added to produce the list length to be finally used.

Of course, these thresholds are also obtained during the training phase, imposing the achievement of a predefined error rate. In this case, the condition may be stronger than the one used above, as we are trying to increase robustness when facing the testing set. In our case we selected also 2%, but measures using 1% will be applied in future tests.

In our nomenclature we will add suffix FX to experiments using fixed thresholds and PP for experiments using proportional ones. So, for example, WN-FX would mean using the winner neuron decision with an added fixed threshold.

Finally, if OPT is added to the method name, it will imply that the threshold used has been chosen for this experiment to achieve less than 2% error rate (it does not give a real measure as the threshold is extracted using the same data we are testing with, but is useful if we consider it gives us an idea of the maximum effort reduction we can achieve with the given method keeping error rate under the predefined value).

4 EXPERIMENTAL SETUP

The experiments have been done on part of the VESTEL database [4] (collected over commercial telephone lines, composed of digits, numbers, commands, city names, etc.). For the initial experiments the training set is composed of 1004 utterances, and the testing and validation sets of 215 words each. This reduced amount of data may lead to inaccurate estimation of the trained parameters, but given the exploratory nature of the work described in this article and the results we are discussing later, it is not a severe problem. Additionally, the total amount of data for training is around 6000 utterances, and they will be used for future experiments.

The task dictionary is composed of 10000 words, so preselection list lengths may vary from 1 to 10000 words.

The evaluation process is as follows:

- The neural network and all the additional system modules are trained using the training database.
- The trained network and system parameters are used in the recognition process, to estimate, for every utterance, the preselection list length to use.
- The inclusion rate is calculated according to these preselection lists as a function of considered candidate.
- The average effort for the task is calculated and compared against the fixed length used in the baseline system.

5 RESULTS

In **Table 2** we present the results of the preliminary experiments carried out on the validation and testing sets, showing both relative reduction in average effort (compared with the fixed list length that would achieve 2% error rate) and the inclusion rate obtained with this reduction and the given method. Only the methods that lead to reduction in average effort are actually shown.

Table 2. Effort reduction and inclusion rate results

	Validation set		Testing set	
Method	Effort reduc.	Inclusion rate	Effort reduc.	Inclusion rate
WN	99.4%	35.05%	99.8%	45.79%
WN-FX	56.0%	95.79%	67.4%	95.79%
WN-PP	51.8%	95.79%	99.3%	66.82%
LC	90.5%	90.51%	90.3%	88.79%
LC-FX	47.0%	96.26%	58.0%	96.26%
LC-FX-OPT	2.9%	98.13%	4.6%	98.13%
LC-PP			34.1%	97.20%
LC-PP-OPT	24%	98.13%	30.2%	98.13%

The experiments using the *winner* (WN) method alone are clearly unable to generate good results (the decision is too *hard*), but any of the other approaches reach reasonable performance with huge decreases in average effort (for example in LC, reductions of up to 90.5% with rates around 90% are achieved). Unfortunately reasonable results are not our objective, and we must stay below the 2% error rate.

The LC-PP-OPT method show that reductions between 24% and 30% could be achieved while keeping error rate under 2%, but, as told before, this is just the optimum. Looking at the actual usable results (those using only the training set to estimate the system parameters), reductions of 34% are obtained while keeping error rate under 3% (LC-PP), and between 47% and 58% for error rates under 4% (LC-FX).

Regarding the different approaches tested, the LC-based methods seem to work better. This may be due to the fact that using a weighted sum allows the system to give longer preselection list than the WN-based ones and this improves robustness in the sense discussed in section 1.3.

6 CONCLUSIONS AND FUTURE WORK

The results shown in this article are preliminary in the sense that there has not been a systematic evaluation process testing the many alternatives we have exposed when implementing a neural network based system.

The fact that there is some empirical evidence of the existence of a relationship between system parameters and the promising results achieved, show that the neural network approach is able to extract the relevant information to some extent.

As stated above, one of the most difficult problems we face is the unavailability of training data able to equally train every output, as this is inherent to the inclusion rate histograms obtained.

Our approach from now on is developing a hierarchical network structure, so that different networks will be used in turn, to finally generate a decision. In the first level, the network will just decide whether the word is supposed to have been recognised in first position or from the second to the last. Additional levels would make further decisions, and the idea is keeping the number of samples to train each output neuron the same, so that a better parameter estimation is expected to be obtained.

Additionally, the relationship between the ANN output and the recognition confidence will be studied, as it is presumably related to preselection list length (the shorter the proposed list is, the higher the confidence in a correct recognition is).

7 REFERENCES

[1] Ferreiros, J., Macías-Guarasa, J., Gallardo, A., Pardo, J.M. and Villarrubia, L. “Recent Work on a Preselection Module for a Flexible Large Vocabulary Speech Recognition System in Telephone Environment”. ICSLP 98. Vol. 2, pp. 321-324. 1998.

[2] Macías-Guarasa, J., Gallardo, A., Ferreiros, J., Pardo, J.M. and Villarrubia, L. "Initial Evaluation of a Preselection Module for a Flexible Large Vocabulary Speech Recognition System in Telephone Environment". ICSLP 96. pp. 1343-1346. 1996.

[3] Villarrubia, L., Gomez, L.H., Elvira, J.M., Torrecilla, J.C. "Context-dependent units for Vocabulary-independent Spanish Speech Recognition". ICASSP 96:. 451-454. 1996.

[4] Tapias, D., Acero, A., Esteve, J., and Torrecilla, J.C. "The VESTEL Telephone Speech Database". ICSLP 94: 1811-1814. 1994