SPANISH RECOGNISER OF CONTINUOUSLY SPELLED NAMES OVER THE TELEPHONE

R. San-Segundo, J. Colás, J. Ferreiros, J. Macías-Guarasa, J. M. Pardo

Grupo de Tecnología del Habla. Departamento de Ingeniería Electrónica. UPM.
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040 Madrid, Spain
lapiz@die.upm.es http://www-gth.die.upm.es

ABSTRACT

In the paper, we present an analysis of the spelling task for Spanish and we describe the research and implementation of a Spanish recogniser for continuously spelled names over the telephone. We analyse and compare three different recognition architectures. The first one is a *Two level architecture*. This approach consists in two steps. In the first one we obtain the most likely letter sequence using the one-pass algorithm. In the second step, to obtain the name recognised, we align the sequence of letters with the different dictionary names using a Dynamic Programming (DP) algorithm. The second alternative consists on an *Integrated Architecture* where a constrained grammar is built with all the names from the dictionary. In this case, we have a higher Name Recognition Rate but the time processing increases a lot. Finally, we propose a combined architecture with a good compromise between recognition rate and time consuming. This approach responds to a strategy of *Hypothesis and Verification.* In the hypothesis stage, we obtain the most likely letter sequence (one-pass algorithm) and then we select N-candidates from the dictionary with a dynamic programming algorithm. In the verification stage, we build a dynamic grammar with the N-candidates and we recognise over it. With this system we obtain a 96.1% Name Recognition Rate in real time for the 1,000 names dictionary and, 92.3% and 89.6% for 5,000 and 10,000 names directories respectively.

Keywords: Spelled names recognition, Spanish spelling task, Recognition over the telephone.

1. INTRODUCTION

Automatic speech recognition of names from theirs spelling is an important sub-task for many applications such as directory assistance [1] or identification of city names for travel services [2]. Natural spelling implies the recognition of connected letters. This is a difficult task, especially over the telephone, because of the confusable letters contained in the alphabet, the distortions introduced by the telephone channel and the variability due to an arbitrary telephone handset.

1.1 The Spelling Task for Spanish.

The performance of a recognition system depends on the size or perplexity of the vocabulary andso on the degree of similarity among the words in the vocabulary. In English, the main difficulty of the spelling recognition task lies in the recognition of the E-set={B, C, D, E, G, P, T, V, Z} [3]. In table 1, we present the transcriptions of the Spanish letters pronunciations using the extended SAMPA Spanish phone set. Looking at table 1, we can identify the E-set for Spanish = {B, C, CH, D, E, G, P, T}.

Spanish Letter transcriptions (SAMPA Spanish Allophone set)									
A	a	**F**	efe	**L**	’ele	**P**	pe	**V**	’uBe
B	be	**G**	ge	**LL**	’eLe	**Q**	ku	**W**	uBe’DoBle
C	Te	**H**	’atSe	**M**	’eme	**R**	’erre	**X**	’ekis
Ch	TSe	**I**	j	**N**	’eNe	**S**	’ese	**Y**	’jGrjeGa
D	de	**J**	’xota	**Ñ**	’eJe	**T**	te	**Z**	’Teta
E	e	**K**	ka	**O**	o	**U**	u		

Table 1: Spanish Letter transcription.

In Spanish, we have to consider another high confusable letters set, the ExE-set = {F, L, LL, M, N, Ñ, R, S}. In this set, all the letters transcriptions have three allophones with the structure **’e _ e**. These letters have only one allophone different (the central allophone), so the acoustics differences (as in the E-set) are minimal.

When we work with continuous speech, another source of recognition errors is the co-articulation between words. This effect is more dangerous when the words of the vocabulary are shorter and theirs pronunciations are similar, as it is in our case. In figure 1, we can see an example of recognition error because the co-articulation effect.

Name Spelled:	R	U	B	E	N
Transcription:	'e rr e	u	b e	e	'e N e
Transcription Recognised:	'e rr e	'u B e		'e N e	
Letter Sequence Recognised:	R	V		N	

Figure 1: Example of recognition error because the co-articulation effect.

In this example, the pronunciations of the letters U and B have been joined to form the letter V, and the letter E was included in the N pronunciation. The letter sequence recognised in this case is quite far from the name spelled.

2. DATABASE

The database used for the experiments is the Spanish SpeechDat version [4]. We have 1,000 phone calls from different persons with a city name, a proper name and a random sequence of letters spelled in each call. The random sequences of letters are used only for HMM training. From the city and property names audio files we have taken randomly 400 for evaluation and 400 for testing, leaving the rest (2200 audio files) for HMM training. We have repeated it three times providing a 3-Round Robin training to verify the results. The results presented are the average of these experiments. We will report the percentage of

Substitutions, Deletions and Insertions, the Letter Accuracy and the Name Accuracy obtained, to evaluate the different alternatives proposed. We consider the *Name Accuracy* as the percentage of cases where the letter sequence recognised matches exactly with the name spelled. All the experiments have been run over a Pentium II 350 Mhz with RAM of 128 Mb, so all the Processing Time results provided are referred to this computer. These results are given in xRealTime units (xRT) as in [5]. 1xRT is the average time spent to pronounce the spelled name.

3. ARCHITECTURES PROPOSED

In all the architectures, we use letter-CHMMs with a number of states proportional to the length of each letter. The shortest model has 9 states and it was associated to the vowel letter I, the longest one has 48 and was associated to letter W. The number of mixtures per state is proportional to the amount of data to train. We consider a minimum number of three mixtures and a maximum of nine. For the speech analysis we use a 10 ms frame shift and a 25 ms analysis window. In the experiments, the static cepstral coefficients, the energy, the first derivative of the energy and the first derivative of the static cepstral coefficients are considered to form the speech parametric representation (a total of 22 coefficients). We use the RASTA-PLP [6] parameterisation as proposed in [7] where we can see a detailed Front-End analysis for the spelling task in English. To compare the different architectures we have considered a 1,000 names dictionary and for the best option we will report the results for 1,000, 5,000 and 10,000 names dictionaries.

3.1 Two Level Architecture.

This architecture consists in two steps. In the first one we obtain the most likely letter sequence using the one-pass algorithm. In the second, to obtain the name recognised, we align the sequence of letters with the different dictionary names using a Dynamic Programming (DP) algorithm accounting for substitutions, insertions and deletions by applying different penalties in each case. This approach is similar to the architecture proposed in [8].

3.1.1 Baseline

In table 2 we can see the results for the baseline experiment:

Results for the baseline experiment.		
One-Pass step	Substitutions	17.8 %
	Insertions	4.2 %
	Deletions	2.8 %
	Letter Accuracy	75.2 %
	Name Accuracy	27.8 %
Whole Recogniser	Name Recognition Rate	89.0 %
	Processing Time	1.2 xRT

Table 2: Results for the baseline system: Percentage of Substitutions, Insertions and Deletions, Letter Accuracy, Name Accuracy and Name Recognition Rate with the Processing Time consumed (considering the 1,000 names dictionary).

3.1.2 Noise Models

In Spanish, there is an important relationship between spelling and pronunciation, so people are not used to spell for clarification and produce a lot of false beginnings, doubts, filled pauses and mistakes. Therefore, on the telephone, the speech input may be contaminated with various ambient noises. To deal with these noises we have included 4 noise models (N-HMMs) in the search space: *[fil]: Filled pause* (filled pause sounds), *[spk]: Speaker noise.* (sounds and noises made by the calling speaker), *[sta]: Stationary noise* (background noise), and *[int]: Intermittent noise* (noises of intermittent nature). The results obtained are presented in table 3.

Incorporating the noise models in the search space.		
One-pass step	Substitutions	16.4 %
	Insertions	1.1 %
	Deletions	2.8 %
	Letter Accuracy	79.7 %
	Name Accuracy	34.3 %
Whole recogniser	Name Recognition Rate	91.5 %
	Processing Time	1.2 xRT

Table 3: Results with the 4 noise models incorporated in search space (1,000 names dictionary).

The incorporation of these models has been very useful for recognition. With almost the same processing time the noise models reduce the substitutions 1.4% and the insertions 3.1% (from 4.2% to 1.1%) obtaining similar results than [3]. The Name Recognition Rate has increased 2.5%.

3.1.3 N-gram Language Models

When the spelled name belongs to a finite known list (dictionary) as in our case, this list can provide very useful information that can be used in several ways [9]. One way of considering this information in the recogniser is defining N-gram language models (LMs) and including them in the search space. We calculate a 2-gram and 3-gram language models using the 1,000 names dictionary (the probabilities were smoothed using a minimum value of probability). Incorporating the 2-gram language model is rather easy because it is not necessary to change the search space. We only need to consider the LM probability when we analyse a possible transition between two letters. In the case of 3-gram LM, it is necessary to change the search space. We have to duplicate every letter model as many times as letters can precede it. One problem we found in this structure was the way of including the noise models in the search space. The solution is to duplicate the noise HMMs as many times as nodes we have in the structure. This solution increases a lot the space size and makes the search very slow. We analysed the noise distribution a long the utterance. We studied the training set and we found that 93.5% of the noises appear at the beginning or at the end of the utterance. So, we decided to incorporate the noise models only in the Initial and End nodes (figure 2). This way, we do not increase the space size so much but we can detect an important amount of noises. As we can see in table 4, the 3-gram LM is more powerful than the 2-gram. We obtain an increment of 7.6% in the Letter Accuracy (because an important reduction in the substitutions) and 25.0% in the Name Accuracy.

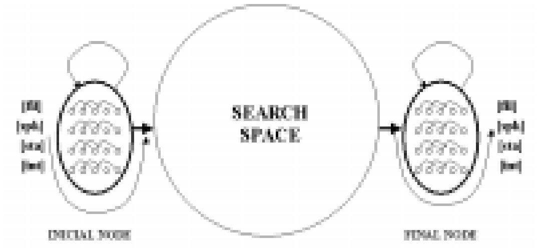

Figure 2: Noise models in the 3-gram LM search space.

	LA (%)	NA (%)	NRR(%)	PT(xRT)
Baseline	75.2	27.8	89.0	1.2
Baseline + N-HMMs	79.1	34.3	91.5	1.2
2-gram + N-HMMs	81.2	35.4	92.1	1.3
3-gram + N-HMMs	89.0	60.4	93.2	3.8

Table 4: Letter Accuracy (LA), Name Accuracy (NA), Name Recognition Rate (NRR) and Processing Time (PT) for the Baseline, Baseline with Noise HMMs (N-HMMs), 2-gram LM with N-HMMs and 3-gram LM with N-HMMs experiments (1,000 names dictionary).

3.2 Integrated Architecture

In this architecture, we use all the names from the dictionary to build a dynamic grammar and then we run the HMM recogniser over this structure. In our case, we build a tree (figure 3) as proposed in [9].

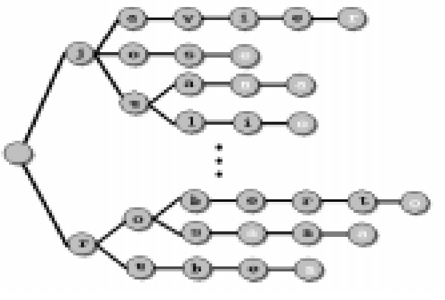

Figure 3: Tree built with all the names from the dictionary.

Including the noise models (N-HMMs) in this structure has the same problems that in the Two Level Architecture when we considered the 3-gram LM. In this case, we have analysed three different possibilities: without noise HMMs, adding noise HMMs at the beginning and end of the search space, and considering the noise HMMs between all possible transitions between letters. The results are presented in table 5.

	NRR(%)	PT(xRT)
Without N-HMMs	94.3	9.2
Initial and End N-HMMs	96.1	9.5
Full N-HMMs integration	96.9	16.7

Table 5: Name Recognition Rate (NRR) and Processing Time (PT) for the Tree structure, considering the three possibilities of including the N-HMMs in the search space (1,000 names dictionary).

Same than in the Two Level architecture, including the noise models permits to increase the recognition rate considerably. In all cases we have higher Name Recognition Rate but the Processing Time increases a lot. We have implemented path-pruning over this structure. In figure 4, we present the results for different thresholds without N-HMMs incorporated.

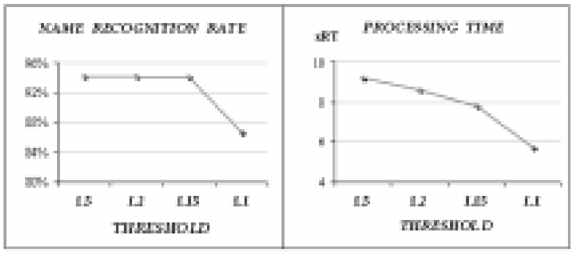

Figure 4: Name Recognition Rates and Processing Times for different thresholds (1,000 names dictionary).

We can no obtain an important reduction in Processing Time without loosing Recognition Rate. In the 1,000 names dictionary case, the tree built has 3,245 nodes (without noise nodes). If we want to consider bigger dictionaries, the Processing Time will increase considerably. This way, we propose a third architecture with a good compromise between recognition rate and time consuming.

3.3 Hypothesis-Verification Architecture

The third architecture proposed is based on a hypothesis-verification approach, similar than [7]. This recognition strategy consists on two steps: in the hypothesis step, we obtain the best letter sequence given acoustics HMMs of the letters and then, we compare it with all the names in the dictionary, using a dynamic programming algorithm. This way we obtain the N-best similar names. These names are passed to the verification step. In this stage, a dynamic grammar is built with the N-best names and the HMM recogniser is rerun with this highly constrained grammar. In figure 5, we can see the block diagram of this architecture.

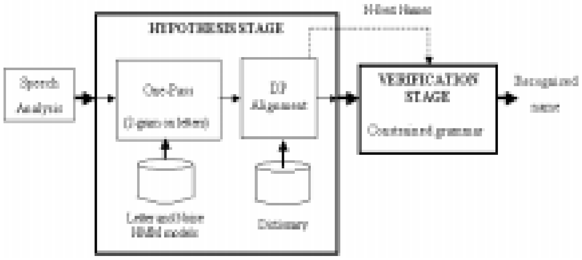

Figure 5: Block Diagram of the Hypothesis-Verification Architecture.

For the hypothesis stage, we have considered the first architecture proposed in section 3.1. We have not incorporated the 3-gram LM because the Processing Time is very high and for the hypothesis stage we need a system as fast as possible. For the dynamic grammar in the verification stage, we have considered a tree, similar to the architecture presented in section 3.2. In this stage, the time consuming is low because the number of names considered is small (N=50 in our case), and we use the same HMMs than in the hypothesis stage, so the state distribution probabilities calculated in the hypothesis stage are stored for the verification stage. In our experiments, we have used the same HMMs for both steps but more detailed models or different recognition parameters can be used in the verification pass. To deal with noises in the verification stage we have tested a fourth alternative. This alternative consists on detecting the noise frames in the hypothesis stage and removing them from the input, before to rerun the recogniser over the N-best name tree. When we detect a noise we do not remove all frames, we keep 5 frames at the beginning and end of the noise segmentation to guarantee that we do not loose speech frames. In table 6, we report the results for the Hypothesis-Verification approach for the different strategies of considering the noise models.

	NRR(%)	PT(xRT)
Without N-HMMs	93.7	2.2
Initial and End N-HMMs	95.5	2.3
Removing noise frames	95.6	2.3
Full N-HMMs integration	96.1	2.9

Table 6: Name Recognition Rate (NRR) and Processing Time (PT) for the Hypothesis-Verification architecture, considering the four possibilities of including the N-HMMs in the verification stage (1,000 names dictionary).

Looking at the results presented in table 6, we decided to keep working with the full noise models integration because we obtain the best Name Recognition Rate with a reasonable Processing Time. With this solution, we have considered several dictionaries of different sizes (1000, 5000 and 10,000 city and proper names) obtained by randomly extracting from the Spanish city and proper name directory. The city and proper names spelled in the database are included in every dictionary. The results for the different dictionaries are presented in table 7.

Size of the dictionary	Hypothesis NRR(%)	Whole System NRR(%)	M	Time (xRT)
1,000 (0.2)	92.1	96.1	50	2.9
5,000 (0.5)	86.7	92.3	50	3.4
10,000 (0.9)	84.2	89.6	50	4.7

Table 7: Name Recognition Rate (NRR) and Processing Time (PT) for the different dictionaries.

The average confusion for the dictionaries is presented into parentheses. These values are a measure of confusion of dictionary [7][10]. In the third dictionary (10,000), there were 9,038 pairs of names that differ only by one letter substitution. This corresponds to an average of 0.9 confusions per name.

4. CONCLUSIONS

In this paper, we present different architectures for continuously spelled name recognition over the telephone. The first approach proposed consists on a Two Level Architecture. In this case, we have analysed the impact of including noise models in the search space. We have demonstrated that including these noise models it is possible to reduce the insertions 3.1%, increasing the Letter Accuracy 4.5% (from 75.2% to 79.7%) and obtaining a 91.5% Name Recognition Rate. Modelling these noises is so important in a Spelled Name Recogniser when the user has no any habit to spell. We have considered the introduction of N-gram LMs (2-gram and 3-gram) in the decoding process. This LM was generated from the directory of the task. We have described an efficient way to consider the noise models (N-HMMs) in the search space generated for the 3-gram LM, obtaining for this case a Letter Accuracy of 89.0% and a Name Recognition Rate of 93.2%. As second alternative, we have considered an Integrated Architecture. In this case, we obtain higher recognition rate but the increment in the Processing Time makes this improvement useless for real time systems. With this architecture we have also proved the importance of considering the noise models in the recognition process. Finally, we propose a combined architecture that responds to a strategy of hypothesis and verification. This architecture has a good compromise between recognition rate and time consuming. For this case we have evaluated the system for 1,000, 5,000 and 10,000 names dictionaries obtaining, 96.1%, 92.3% and 89.6% Name Recognition Rate respectively. A Real-time version of this architecture has been implemented on a Pentium III 600 Mhz with 256 Mb of RAM, working over the telephone network.

5. ACKNOWLEDGMENTS

The authors are very grateful to Rubén Pacheco for his help to implement the algorithm for training the HMMs.

6. REFERENCES

1. Lehtinen. G., et al., 2000. “IDAS : Interactive Directory Assistance Service”. VOTS-2000 Workshop, Belgium.
2. Ward W., Pellom B. “The CU Communicator System” Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Keystone Colorado, 1999.
3. Loizou. P. and Spanias. A. “High performance alphabet recognition” IEEE Trans. On Speech and Audio Processing, Vol. 4, No. 6. 1996
4. Moreno, A. *SpeechDat* [cd-rom]. Ver. 1.0. [Barcelona]: Universitat Politècnica de Catalunya <http://www.upc.es/castella/recerca/recerca.htm>, c1997. 4 cd-roms. (Spanish Fixed Network Speech Corpus).
5. Ravishankar, M.K. “Efficient Algorithms for Speech Recognition”. Unpublished PhD Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 15, 1996.
6. Hermansky, H., Morgan. N., Bayya A., Kohn. P. “Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP)”. in Proc. EUROSPEECH. pp. 1367-1370. 1991
7. Junqua, J.C. “SmarTspelL^{TM}: A Multipass Recognition System for Name Retrieval over the Telephone”. IEEE Trans. On Speech and Audio Processing, Vol. 5, No. 2, March, 1997.
8. Jouvet. D., Lainé. A., Monné. J., and Gagnoulet. C., “Speaker independent spelling recognition over the telephone”. in Proc. ICASSP, pp. II.235-II.238. 1993
9. Hild. H., and Waibel. A., “Recognition of spelled names over the telephone”. in Proc. EUROSPEECH. pp 346-349, 1997.
10. Cole, R.A., M. Fanty, M., Gopalakrishnan, M. and Janssen, R.D.T. “Speaker-independent name retrieval from spellings using a data base of 50,000 name” in Proc ICASSP, pp 325-328, 1991.