USING NON-WORD LEXICAL UNITS IN AUTOMATIC SPEECH UNDERSTANDING

M. Peñagarikano, G. Bordel, A. Varona, K. López de Ipiña

Dpto. Electricidad y Electrónica
Universidad del País Vasco (UPV/EHU)
Lejona, Vizcaya, SPAIN

ABSTRACT

If the objective of a Continuous Automatic Speech Understanding system is not a speech-to-text translation, words are not strictly needed, and then the use of alternative lexical units (LUs) will bring us a new degree of freedom to improve the system performance. Consequently, we experimentally explore some methods to automatically extract a set of LUs from a Spanish training corpus and verify that the system can be improved in two ways: reducing the computational costs and increasing the recognition rates. Moreover, preliminary results point out that, even if the system target is a speech-to-text translation, using non-word units and post-processing the output to produce the corresponding word chain outperforms the word based system.*

1. INTRODUCTION

The use of words as lexical units (LUs) in Continuous Automatic Speech Understanding has been, basically, not questioned. Only very few recent papers deal in some way with alternative units [1][2][3][4]. The need for new units has been better seen from languages were the word concept is no clear (i.e. Chinese) or those were words are highly structured (i.e. German or, to a lesser extent, Spanish). We thought that the only reason to adopt the word is given by the fact that normally it is the element composing texts, and then word based Language Models can be learnt straightforwardly. Our interest for alternative LUs came from this field: Language Modelling. Using words as units, our models ignore important internal word structure and phrasal structures are modelised with a high cost. If the system objective is not a speech-to-text translation, words are not strictly needed, and then the use of alternative LUs will bring us a new degree of freedom to improve the system performance.

Once the word is questioned, many approaches can be adopted to investigate the alternatives. As the word is used as a connecting element between the phonetic knowledge and the syntactic-semantic-pragmatic knowledge (the Language Model), an adequate adaptation to both parts would lead to the best results. Nevertheless, the high computational costs suggest that we should solve the problem only partially, trying to obtain a performance improvement for the language model and observing if it results in an improvement of the whole system.

Next section describes different aspects we tested to automatically obtain LU sets from samples. The experiments carried out gave us some results that are briefly shown in section 3. The obtaining of the LUs was carried out on textual information attending to the perplexity given by the Language Model (3.1), whereas the whole system evaluation is made in terms of recognition rates (3.2).

2. OBTAINING THE ALTERNATIVE LEXICAL UNITS

To automatically obtain a set of LU from a database we propose a procedure based on the alteration of a predetermined set. Two algorithms are tried. The first one needs a criterion to generate the new units, and the second needs a criterion to evaluate the performance given by the altered sets. So, the following paragraphs are devoted to these four aspects:

- The Initial set of units.
- The algorithms.
- The generation of new units criterion.
- The evaluation criterion.

2.1 Initial set of units

The computational cost of the analysis for one utterance is linear to its length and at least quadratic to the number of LUs considered by the system (that is the case for a smoothed bigram model). So, if it were not that the recognition rates drop dramatically (for a fixed kind of LM), it would be worth using single phonemes as LUs because of the reduction of the quadratic dimension at the expense of the linear one. Obviously, the low recognition rate is due to the loss of the information about phoneme combinations contained in the word set.

So, we can start our search for a good LU set with the phoneme set. As the new units were generated, the recognition rate and the computational costs will grow. The hope is that at some point the performance will be more satisfactory that the word based system.

As we will see later, the criterion for new word generation is based on probabilistic considerations on the database. This fact made us realise that those words appearing only few times in the database had no chance to be formed from phonemes. So, three alternative initial sets were also tried: phonemes plus words appearing three or less times, twice, and once.

Two more initial sets have been tried: supposing that semantic constrains are good criteria to form LUs, we also tried a set of pseudo-morphemes (not exactly morphemes in the linguistic sense but a very close approximation), and attending to the idea that recognition rates can be improved at the expense of increasing the computational cost, we also used the word set as an initial set.

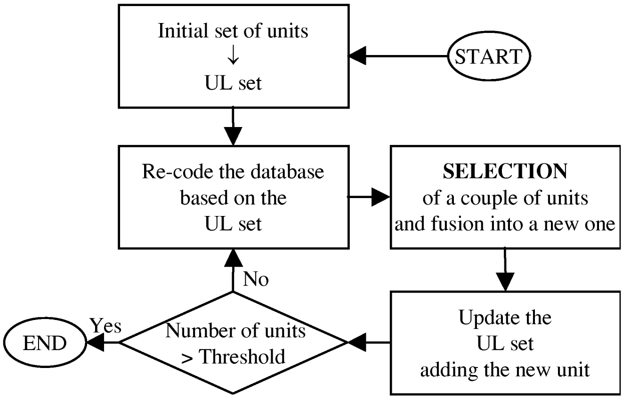
* This work has been partially supported by the Spanish CICYT under grant TIC-94-0210-E and by the UPV/EHU under grant UPV-224.310-EA036/97.

As a result, these are the initial sets tried:

- Phonemes
- Phonemes + words appearing once.
- Phonemes + words appearing once or twice.
- Phonemes + words appearing once, twice or three times
- Pseudo-Morphemes
- Words

2.2 Algorithms

Two algorithms had been applied. The first one implements a simple *Greedy* scheme consisting on the iterative generation of new LUs according to the selected criterion (see **Figure 1**). The success of this mechanism relies entirely on the appropriateness of the generation criterion.



Initial set of units ↓ UL set
START
Re-code the database based on the UL set
SELECTION of a couple of units and fusion into a new one
Update the UL set adding the new unit
Number of units > Threshold
No
Yes
END

Figure 1. The first algorithm to obtain the LU sets uses a *Greedy* approach.

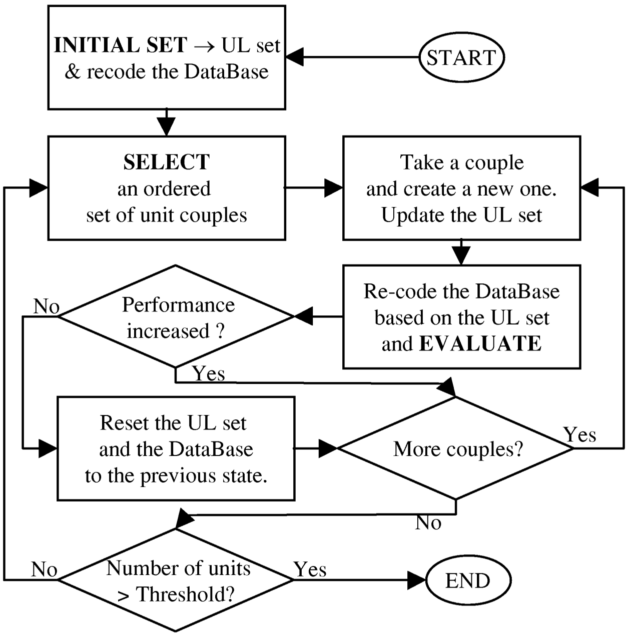
The second algorithm is taken from [6], where it is used to obtain phrasal structures. This algorithm tries a more intelligent evolution of the LU set assuring that the recognition rate monotonically decreases all through the execution. This is implemented as a *Local Search* scheme. The optimal implementation consists on trying, at each step, all the possible new LUs and selecting the one showing the highest reduction of the recognition rate. This strategy is computationally prohibitive, so a sub-optimal approach is implemented: a subset is determined, and iterativelly all the units improving the performance are accepted before a new subset is formed. The construction of these ordered subsets is based on the same criteria used in algorithm 1.

2.3 Generation of new units criteria

The new units are always generated by concatenation of two previously existing units. The selection of the two units to be joined is based on the maximisation of a predetermined function. Three functions were chosen to be tested. The simplest one is the frequency observed in the database. This approach was also adopted in [1]:

$$\mathbf{F}(u,v)=\frac{\mathbf{N}(u,v)}{N}$$

were u and v are the LUs, **F** the frequency function, **N** the count function and N the total number of LUs composing the sentences of the database.



INITIAL SET → UL set & recode the DataBase
START
SELECT an ordered set of unit couples
Take a couple and create a new one. Update the UL set
Re-code the DataBase based on the UL set and EVALUATE
Performance increased ?
No
Yes
Reset the UL set and the DataBase to the previous state.
More couples?
Yes
No
Number of units > Threshold?
No
Yes
END

Figure 2. The second algorithm to obtain the LU sets uses a “sub-optimal” *Local Search* approach.

A second function we tried was a correlation coefficient (**CC**) as it is proposed in [6] to select word phrases in a recognition task:

$$\mathbf{CC}(u,v)=\frac{\mathbf{N}(u,v)}{\mathbf{N}(u)+\mathbf{N}(v)}$$

Observing that low frequency LUs can present high **CC** values but they have low impact in the final performance of the system, we defined a modified CC function (**MCC**) diminishing these values:

$$\mathbf{MCC}(u,v)=\mathbf{CC}(u,v)\ \mathbf{N}(u,v)=\frac{\mathbf{N}(u,v)^2}{\mathbf{N}(u)+\mathbf{N}(v)}$$

2.4 Evaluation criteria

In this work, we tried to better the recognition system performance by improving the Language Model for a fixed phonetic model. Hence, for Algorithm II we used a LM evaluation: the perplexity. A more realistic whole-system evaluation would be computationally unaffordable. Nevertheless, the UL sets obtained were evaluated via recognition rate of the whole system.

The Perplexity function as usually expressed to comparatively evaluate Language Models is not valid in this case. There is a dependency on the units used to compose the sentences, so it must be altered to be invariable to the units change. This can be straightforwardly accomplished by basing the evaluation on an invariable unit, in our case the phoneme:

$$\mathbf{PP}(M)\Big|_{T}=2$$

```
ERROR: undefined
OFFENDING COMMAND: ‰  ,oøk-    Cœ

STACK:

-dictionary-
```