NaniBD: a Set of Tools for Transcribing and Validating Speech Databases

Albino Nogueiras Rodríguez & Asunción Moreno Bilbao
Speech Processing Group - GPS/TSC
Universitat Politècnica de Catalunya
Barcelona, SPAIN
[albino,asuncion]@gps.tsc.upc.es

Abstract

This paper describes NaniBD, a set of tools designed for transcribing and validating speech databases, developed at the Signal Processing Group (GPS) of the Department of Signal Theory and Communications of the Polytechnic University of Catalonia (TSC/UPC). The main purpose of its development was the need of a revision system in order to validate and annotate the Spanish corpus of SpeechDat (II) in the speech processing environment available at GPS. Despite of this, NaniBD is designed as a general-purpose system that might fit any other database, idiom or speech processing system. So far, the system has been used to revise some 200,000 speech files from three different corpora. In this paper we will focus our attention to the actual implementation used in the transcription of a SpeechDat (II) specifications compatible Catalonian corpus. 1000 speakers, each of them uttering 44 files, compose this corpus. In this application, we use speech-noise detection, automatic recognition of spontaneous prompts, digit and letter to text translation and access to an external database in order to minimise the amount of time spent by human operators in the revision procedure.

Fundamentals of NaniBD

The transcription and validation is one of the most important tasks in speech databases acquisition. It can be said that the value of a speech database relies greatly in the accuracy of the transcription of its contents. The final objectives of the transcription process may vary a lot for different tasks, but they usually involve the audition of each signal file, the introduction or validation of its orthographic contents and the annotation of any other possible event such as noises, mispronunciations and so. In certain circumstances it may be necessary to perform any other action. For instance, the acquisition strategy used in SpeechDat (II) implies that nothing is known about the speaker until his files are listened to. So we need to record some important data about him such as his gender, age, etc. during this phase. There are many different actions that may be interesting to perform in order to minimise the amounts of work and time spent by human operators during the validation process. NaniBD has been designed with four main objectives in mind: robustness, flexibility, ease of use and speed.

- The revision system must be robust in the sense that it must *work by itself*. That means that, once configured and started the procedure there must be no doubt about each file of each speaker will be revised once and only once. No matter how long it takes to revise the whole database, either one hour or one year; how many revisers work concurrently; or any other thing. In order to achieve a system as robust, in the above sense, as possible, NaniBD performs **Speaker and File Selection** and **Speaker Locking** which are explained below.
- It must be flexible in the sense that there are many different possible actions to be performed that may produce a reduction in the reviser's load. And these actions may be completely different in nature (digit translation and automatic speech recognition, for example). NaniBD implements **External Command Execution**, also explained below, as a means of achieving this flexibility. It consists in the invocation of different shell scripts called from within the different tools of NaniBD with suitable arguments and at certain instants.
- In order to make the system as fast and easy of use as possible, two different tools have been developed: a general-purpose speech databases processing program, **ProcBD**; and a graphical, ncurses based, revision tool **RevBD**. The first is intended to perform all those actions that may be done before or after the revision phase. The second is the one with which the reviser actually interacts. Its design has been focused to minimising the interaction between the reviser and the system.

In the following of the paper all these issues will be presented along with an explanation of a real and quite complete application of the system: the validation of the Catalonian corpus

Speaker and File Selection

Most speech databases present a tree directory structure that reflects its contents in terms of speakers and/or sessions. NaniBD assumes that the structure of the database is as follows:

- The whole structure is mounted in a parent directory that may be different as a function of the type of file (for example: /veu/senyal/BD3/cta could be used for storing Catalonian A-law files, .cta, while /veu/senyal/BD3/cto could store the Catalonian orthographic files .cto).
- All the files of a single speaker are held in one sub-directory of the parent directory, and this directory only holds data of this speaker. The path of the directory where the data of one speaker is held may include sub-directories itself (for example: disk2/ses3/4069tf).

Different types of data do have different three letter extensions (plus a leading point) and may or not be held in different parent directories, but the sub-path of the speaker and the name of the file, without the three letter extension, are always the same.

For example, table 1 could be the names for different files of two speakers, 4069tf and 5312vn, and for three different types of file (in the GPS system the extensions .cta, .cto and .prm, correspond to the recorded signal, the orthographic contents and the parameterised data). Note that, in this case, both .cta and .cto files share the same parent root, while .prm files are stored separately.

/veu/Sen/	disk2/4069tf/	4069I1.cta
		4069I1.cto
		4069P2.cta
		4069P2.cto
	disk4/5312vn/	5312I1.cta
		5312I1.cto
/veu/Prm/	disk2/4069tf/	4069I1.prm
		4069P2.prm
	disk4/5312vn/	5312I1.prm

Table 1: Directory Structure

Whenever a structure like this applies, a speaker by speaker basis in the scheduling of the revision tasks will be convenient and, probably, desired.

NaniBD allows two ways of specifying the names of the speakers and files to be revised: by means of a list or using masks for both the speakers and their files. When using a list it is supposed to be an ASCII file with one line specifying the filename of each file to be revised without either the parent directory or the extension. In the case of the above example, a file with the complete list would look as follows:

```
disk2/4069tf/4069I1
disk2/4069tf/4069P2
disk4/5312vn/5312I1
```

Figure 1: Sample of List

Notice that, to form the complete filenames, a parent directory and a trailing extension must be explicitly given or assumed.

The other and more useful way of specifying the names of speakers and files is using a filename creation mask in the same usual way of UNIX like systems. Two different masks are used:

- MaskLoc is the mask of the speakers and is supposed to point to a subdirectory of the parent directory, DirSen.
- MaskSen is the mask of the different files of each speaker and is supposed to, in conjunction with the extension of the signal files, ExtSen, point to each of the files.

In this way, we could express the names of the speakers and files of the example as Table 2 shows.

DirSen	/veu/senyal/BD3/SpeechDat/Sen
MaskLoc	disk[0-9]/[0-9]*[a-z]
MaskSen	*{I1,P2}
ExtSen	cta
fullname	$DirSen/disk[0-9]/[0-9]*[a-z]/*{I1,P2}.cta

Table 2: Filename Generation from Masks

The filename creation mask convention used is very similar to that of UNIX `/bin/ls`. Actually, the programs do call `/bin/ls` in order to determine the speakers and files to be processed. Nevertheless, unlike `/bin/ls`, in NaniBD duplicates are eliminated. By doing so we can manage to schedule the speakers and files revision in the order we like without duplicating work. For instance, in the example above, if we wished to process first the two files per speaker indicated, and all the rest of the files in the directory afterwards, the new MaskSen should be *{I1,P2,*} The main advantage of this approach is that we no longer need to know if there are new files to revise, we will always schedule to revise any thing below the parent directory that conform to the masks given to the speakers and files.

Speaker Locking

No matter which method is employed in naming the files to be revised, we must ensure that each file is revised once and just once. In order to accomplish this, NaniBD implements a speaker locking strategy. The objectives of this strategy are:

- revising or processing only those speakers prepared to be;
- not revising or processing any speaker whose revision or process has already started;
- and keeping track of the speakers successfully revised or processed and of those who failed.

This is done using three different locks:

- ExtBlqPre: extensión previa de bloqueo (previous locking extension). If this is not null, only those speakers such that a file with the same name as the speaker and with a trailing extension equal to .ExtBlqPre exists in the parent directory of the source data, will be processed.
- ExtBlqIni: extensión de bloqueo inicial (initial locking extension). If this is not null, only those speakers such that no file with the same name as the speaker and with a trailing extension equal to .ExtBlqIni exists in the parent directory of the target data, will be processed. Also, each time the processing of a new speaker is started this file is created, so it prevents another process to start it also.
- ExtBlqFin: extensión de bloqueo final (final locking extension). If this is not null, only those speakers such that no file with the same name as the speaker and with a trailing extension equal to .ExtBlqFin exists in the parent directory of the target data, will be processed. Also, each time the processing of a speaker is finished this file is created, so it prevents another process to start it also. If a file with the extension .ExtBlqIni exists, it is deleted after the former is created. That means that speakers that cannot finish correctly their

process will present the initial locking file but not the final, so it is possible to keep track of them.

Using both the initial and final locking extensions ensures that all the speakers will be processed only once in no matter which condition. In this way, for example, we can add revisers at any time. Also, in combination with File and Speaker Selection, it allows the possibility of scheduling routine tasks to be done every night out of office hours.

External Command Execution

In order to make NaniBD extensible, portable, and compatible with any speech processing system (HTK, Ramses, Matlab, etc.) an external command execution strategy has been chosen. This strategy is based on the execution of commands defined by the user and that are called with suitable arguments at certain points in both the processing or revision of a speech database. The number of arguments is always four, being the first three fixed: the parent directory of the speech signal files, DirSen; the parent directory of the orthographic label files, DirMar; and the parent directory of the resulting files, DirRes. The fourth argument varies as a function of the nature of the external command. Two kinds are considered: file by file and speaker by speaker commands. A file by file commands is executed once per file and its fourth argument is the name of the file, NomSen. A speaker by speaker command is executed once per speaker and its fourth command is a list, as those seen in Speaker and File Selection, with the name of all the files of the speaker, LisSen.

Orthographic Label Files

The main purpose of the revision and validation procedures is to guarantee that each speech file has a companion file with an accurate orthographic transcription of its contents. The orthographic transcription along with other data related to the file, the speaker, the recording conditions or even the revision session itself are stored in the orthographic label files. Only RevBD operates directly on their contents. Any other operation not considered by RevBD should be carried out using the external commands of ProcBD or RevBD.

The following table lists the labels RevBD recognises and operates on. Other labels will be silently ignored:

- `LBO`: if this label exists, its fourth datum is the orthographic transcription of the recorded phrase. The results of the revision will always be written in this field. If it existed in the input file, its contents will be taken as the initial value for the revision procedure.
- `LBR`: if this label exists, its sixth datum is the theoretical orthographic contents, in the case of textual phrases, or the prompt, in the case of spontaneous answers. If it existed in the input file and no `LBO` label could be found, its contents will be taken as the initial value for the revision procedure.
- `SAT`: the only datum of this label is the fraction (over one) of clipped samples of the speech file. If this field exists, its value (expressed in terms of parts per million, PPM) will be printed by RevBD. If its value raises over 50PPM it will be highlighted.
- `SNR`: its only datum is the signal to noise ratio expressed in dB's. If this file exists, RevBD will print its value. If its value falls below 20dB, it will be highlighted.
- `LBE`: if this field exists, its two data indicate the (probably automatically detected) initial and final samples of speech in the file. Its values are employed by RevBD in order to avoid listening to segments known to contain nothing but silence. Their values may be independently annulled and restored from within RevBD.
- `EXP`: this label is created, or overrode in the case it existed previously, by RevBD. Its only datum indicates the login name of the reviser.
- CRS: if this field exists, its only datum includes the signal comment.
- `DAT`: this label is created, or overrode in the case it existed previously, by RevBD. Its only datum indicates the date and hour when the file was revised.

Speaker Files

There are several data about the speaker or the recording conditions that should affect equally all the files of each speaker. Although **SpeechDat (II)** specifications tell to include these data in every orthographic label file, **NaniBD** uses one single ASCII file per information and speaker. The name of the files where these data are held is formed in a similar way to that employed in forming the locking files: the name of the speaker followed by a point and a three-letter extension. The different data managed, the corresponding extensions and possible contents are listed in the following scheme:

- **Speaker's Identification Number**. Extension: **`.NId`**. It contains one single line with four o six digits in it. If four digits are used, they are supposed to be the speaker's identification number. If six digits are used, the trailing two digits are control ones that can be used to detect and correct possible errors. It should be noted that, many times, the identification number is needed in order to know the theoretical contents of the speaker's files, so no further processing can be done until it is determined.
- **Speaker's Sex**. Extension: **`.Sex`**. It contains one single line with the gender of the speaker according to the codes indicated in the Sex file. In our case it includes one for male, one for female and another for ambiguous.
- **Speaker's Birth**. Extension: **`.Nat`**. It contains one single line with the speaker's year of birth (two last digits) in it.
- **Speaker's Dialect** Extension:**`.Dia.`**It is similar to the **Speaker's Sex** file but using codes adapted to the main dialectical variants of Catalonian: *Central*, *Nordoccidental*, *Balear* and *Valencià*.
- **Recording Conditions**. Extension: **`.Amb`**. It is similar to the **Speaker's Sex** file but using codes adapted to the typical environments found in the recordings.: ambiguous, particular home, public place, telephone box, or mobile telephone.

ProcBD: a General Purpose Speech Databases Processing Tool

Many times it is useful to apply any kind of processing to the speech or orthographic data prior to or after of its revision. One such useful processing could be voice/silence detection in order to alleviate the total amount of data to revise by not doing so in those segments known to hold nothing but noise. Another one could be the automatic recognition of the spontaneous prompts, etc. ProcBD provides a way to perform such tasks with little supervision and adapted to the needs and characteristics of the database to be revised. As a matter of fact, ProcBD, by its own, does nothing. Yet, with suitable locking extensions, it would do nothing but, at less, it would do it only once. In order to do anything of profit, ProcBD, relies on the use of external commands. These commands are executed over the speakers and files selected according to Speaker and File Selection and Speaker Locking. Three different commands are possible:

- CmdPreLoc: is executed every time the processing of a new speaker's files is started.
- CmdSen: is executed for every file of each speaker.
- CmdPosLoc: is executed every time the processing of a speaker's files is finished.

All three commands are called with four arguments. The first three ones are the parent directories of the signal files the orthographic files and where the processed files are to be left, respectively. The fourth is the name of the file, in the case of CmdSen, and a list with the names of all the files of the speaker, in the case of CmdPreLoc and CmdPosLoc.

RevBD: a Graphical Tool for Transcribing and Validating Speech Databases

RevBD is the main revision tool of NaniBD. It works on a graphical environment based on ncurses and its design is directly aimed to minimising the interaction between the reviser and the system. The goal is that, if no correction is to be done, just one keystroke (the enter key) suffices to save the current file, and see the next file contents while listening to it. Besides, when interaction is needed, it has been minimised by using only single keystrokes or control keys whose meanings are more or less explained at screen using almost pneumonic words that reduce greatly the training time of the revisers. Yet, from the revisers point of view, it is quite a touch-and-feel program (Spanish revisers should be said, all information is written in Spanish).

Most of the features of RevBD are similar to those of ProcBD but adapted to interactive operation. Thus, RevBD implements speaker and file selection, speaker locking and external command execution. Besides, it also allows listening to the signal file, editing its orthographic contents and/or speaker data and executes External Commands at will.

RevBD Windows

On start-up, RevBD prints a screen divided in fiven windows surrounded by boxes (see Table 3); plays the file; places the cursor on the second of the windows; and waits for input from the user. The five windows are, from top to bottom: the file information window, the main edition window, the speaker and file comment edition windows and the help window. Just above the main editing window there is the status line, which provides information about the current situation.

The File Information Window. This window shows information about the speaker and the file under revision.

- Locutor (Speaker): shows the name of the speaker.
- Frase (Phrase): shows the name of the file.
- No. Iden. (Identification Number): shows the identification number of the speaker. If it is not known prior to the validation process, a string of six question marks, ?????? , is printed instead.
- Nacimie. (Birth year): shows the birth year of the speaker. If it is not known prior to the validation process, a string of four question marks, ???? , is printed instead.
- Sexo (Sex): shows the gender of the speaker. If it is not known prior to the validation process, a string of six question marks, ?????? , is printed instead.
- Ambiente (environment): shows the environment or recording conditions of the speaker's recording session. If it is not known prior to the validation process, a string of six question marks, ?????? , is printed instead.
- Dialecto (dialect): shows the dialect of the speaker. If it is not known prior to the validation process, a string of six question marks, ?????? , is printed instead.
- Volumen (volume): shows the amplification or attenuation applied to the file during its reproduction.
- SNR (signal to noise ratio): shows the SNR of the file. If it is not known prior to the validation process, 0dB is printed instead. If its value falls below 20db, it is highlighted.
- Satura. (Saturation): shows the fraction, in parts per million, of samples of the file which have been clipped. If it is not known prior to the validation process, 0PPM is printed instead. If its value raises above 50PPM, it is highlighted.
- Duracion (duration): shows the duration in second of the file.
- Inicio (start): shows the detected start of the speech in the file. If no silence-speech detection was performed (no LBE label could be found in the input file), or its value had been voided, 0.0s (the absolute start of the file) will be printed instead.
- Final (end): shows the duration in second of the file. If no silence-speech detection was performed (no LBE label could be found in the input file), or its value had been voided, the absolute end of the file will be printed instead.

Speaker and File Comment Edition Windows. This two one-line windows show the contents of the speaker and file comments. These comments may be edited by pressing keystrokes Ctrl-U and Ctrl-O respectively.

```
Locutor : test/dr1/mdab0  Nacimie.: ????      Volumen: +0.0dB   Duracion: 2.1s
Frase   : si1669          Sexo     : Femenino  SNR    : 46.8dB   Inicio  : 0.0s
                          Ambiente: Domicilio Satura.:  0.0PPM  Final   : 2.1s
No.Iden.: ??????          Dialecto: ???????
Insertar
Be excited and don't identify yourself.



Comentario Locutor   [                                                    ]

Comentario Senhal    [                                                    ]

continuar          Intro   Insertar/sobre      Insert   mala pronunciacion *
fichero siguiente Av.Pag   palabra siguiente   Ctrl-W   ininteligible     **
fichero anterior  Re.Pag   palabra anterior    Ctrl-B   pausa rellena (mm) @
locutor anterior      F1   Inicio/Fin      Inicio/Fin   ruido de locutor   ^
locutor siguiente     F2   borrar texto            F5   ruido estacionario &
                           Limpiar pantalla    Ctrl-L   ruido impulsivo    $
Guardar fichero   Ctrl-G                                truncado hardware  =
Recuperar fichero Ctrl-R   Cantar fichero      Ctrl-C
EJecutar demanda  Ctrl-J   Subir volumen       Ctrl-S   Num. identif. Ctrl-N
Finalizar         Ctrl-Q   Disminuir volumen   Ctrl-D   Sexo locutor  Ctrl-X
                           altavoZ/auricular   Ctrl-Z   fEcha nacim.  Ctrl-E
Ambas marKas      Ctrl-K                                Ambiente grab Ctrl-A
Marca Inicio      Ctrl-I   comentario locUt.   Ctrl-U   dialecTo loc. Ctrl-T
Marca Final       Ctrl-F   cOmentario frase    Ctrl-O
```

Table 3: RevBD Main Windows

The Help Window. This is the bottom window and the biggest of them all. Its only function is to be a reminder of available keys and their function. At this time, all information is written in almost Spanish.

RevBD Main Operation

As was mentioned before, normal execution of RevBD starts by playing the fist file to revise (if none is available, a message telling so will be printed and the program will exit), writing its contents in the main edition window and waiting the user for input at the beginning of this window. The user may input four kinds of characters:

- Normal characters are those that are allowed to be written in the orthographic label file. They are:
    - Lowercase letters, spaces and punctuation marks: all these characters are written directly at the point of the cursor according to the insertion/overstrike mode.
    - Annotation characters: they are equivalent to those specified by SpeechDat (II) but abbreviated to just one or two characters in order to reduce input from the user. They appear highlighted in the main edition window.

*	Bad pronunciation
**	Unintelligible
@	Filled pause (mm)
^	Speaker noise
&	Stationary noise
$	Impulsive noise
=	Truncated by hardware

- Digits and uppercase letters: these characters should not be left in the file, but may be useful allowing them during the revision in order to make it easier. These characters are highlighted during the revision and, in the case that any such one would be left in the file after revision, the reviser is alerted of this fact and asked for confirmation before leaving the file.
- Edition keys: are those used during the edition not meant for being displayed. They are the usual edition keys and other ones whose meanings are explained in the Help Window.
- Command Keys
    - File command keys. The most important is the Enter key, which selects the next file to be revised, and plays its contents. Other keys allow go faster through the files and speakers, recover the original version of the file and save the current file.
    - File playing keys. They involve playing the current file and increasing or decreasing the volume.
    - Start/End marks control keys, They allow the deletion or restoration of the marks of beginning and end of speech. They are useful when the reviser detects truncation of speech, because it can be produce by bad speech/silence detection.
    - Special command keys

Ctrl-J	run command specified by -d option
Ctrl-N	identification number edition mode
Ctrl-X	speaker sex selection mode

Ctrl-E speaker’s year of birth edition mode
Ctrl-A environment selection mode
Ctrl-T dialect selection mode

- Other control keys

Ctrl-Q quit the program
Ctrl-L refresh the screen.

If a key not included in this table is pressed, a beep will sound and nothing will be entered. If everything in the file is correct, the reviser only has to press the Enter key and the following steps will be executed:

1. The current contents of the file are written on the target directory.
2. If the file had been modified and the option -M CmdModSen was used, the command will be executed taking as fourth argument the name of the current file.
3. If the option -S CmdPosSen was used, the command will be executed taking as fourth argument the name of the current file.
4. If the option -s CmdPreSen was used, the command will be executed taking as fourth argument the name of the next file.
5. The next file is loaded and its contents played on the selected device.

If the reviser needs to go faster to a determined file, he can use the PgUp or PgDown keys in order to load the previous or next file respectively without listening to its contents. When the last file of a speaker is reached, the reviser is asked if the speaker is to be passed or failed. If the speaker is to be considered as passed, the external command indicated by the option -L CmdPosLoc is executed. Its fourth argument will be the list of files of the speaker. This command is executed just after any individual command is applied to the last file of the speaker. Then, the external command indicated by the option -l CmdPreLoc is executed taking as fourth argument the list of files of the next speaker and his first file is loaded. The reviser may also go directly to the first file of the next/previous speaker by using the F1 and F2 keys. Note that if this option is employed, any file not passed through will not be considered as revised and nothing will be written in the target directory. A file will and only will be considered revised if its contents are displayed on the screen, even if it is never listen to.

During the revision of a speaker’s files, the user may have to check or enter the information about the session or the speaker. The data RevBD uses are the identification number, the speaker’s sex, year of birth and dialect, and the recording environment. NaniBD allows a special treatment of these data by highlighting its absence and warning the reviser before allowing him to leave one speaker’s files without having defined the corresponding information. In order to modify these values, it is necessary to enter special edition or selection modes.

Finally, the use of the **External Command under Demand** will allow the reviser to perform any previously defined action at any time he wants..

Identification Number Edition

Pressing Ctrl-N enters the identification number edition window. This window appears in the middle of the screen and allows the user to enter a number of four or six digits. Its fist mandatory four digits form the actual identification number. The trailer two digits are used as control digits. The first of them is the rest of the integer division of the sum of digits of the number divided by ten. The last one is the rest of the integer division of the sum of each digit multiplied by its index divided ten. These two control digits not only allow the detection of possible errors but may correct up to one incorrect digit as well. If a six-digit number is entered and its control digits do not agree, a beep will be sound and the number will not be entered. If this control is not desired, entering just four digits suffice because the trailing two will be automatically added.

To enter the number and return to the main edition mode, the user must press the Enter key. If the identification number has been modified and the option -I CmdNumId was used, CmdNumId will be executed taking as its fourth argument the name of a file with the list of files of the speaker.

Option -T may be used in order to start each new speaker first file in identification number edition mode.

Speaker’s Sex and Dialect and Environment Selection

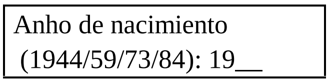
Pressing Ctrl-X enters the speaker’s sex selection window, Ctrl-T enters de speaker’s dialect selection window and Ctrl-A the environment selection window. These three windows appear in the middle of the screen and both of them ask the user to enter a key. The contents of the window depend on the kind of information and are customisable through external files indicated in the options given to RevBD. The structure of these files is quite simple: an ASCII file with one line per option.

In any of the speaker’s data selection windows, the reviser may enter four kinds of keys:

- a digit from one to the number of options, in which case the option with this digit is selected;
- a letter that must coincide with the first letter of any of the options, in which case the first option that starts, case insensitively, with this letter is selected;
- the enter key, which selects the currently highlighted option;
- or an up or down arrow, which selects the following or previous option.

Speaker’s Year of Birth Edition

Pressing Ctrl-E enters the speaker’s year of birth edition window, which appears in the middle of the screen with the following message



Anho de nacimiento
(1944/59/73/84): 19__

Figure 2: Speaker’s Year of Birth Edition Window

and waits until the user enters the two last digits of the speaker’s year of birth. The proposed years correspond to the standard ages adopted by SpeechDat (II) specifications.

External Command under Demand

The option -d CmdSenDem allows the user to specify an external command that will be executed with the name of the current file as fourth argument each time the reviser decides to do so. Before the command is executed, the current file contents are saved; and, after the command has exited, the, eventually modified, file is loaded from the target directory. This option allows the user to enter digits, letters or abbreviations by specifying a convenient translator as external command and calling it each time such input is used. When the command is executed a window appears in the middle of the screen informing of this fact, if the command outputs nothing either by standard or error outputs, this window will silently disappear and execution will continue normally. In the case the command produces any output, a message asking for a key to be pressed appears and execution is locked until it is done.

Catalonian Corpus Revision Procedure Overview

The revision of the Catalonian corpus designed to be compliant with SpeechDat (II) specifications has enabled us to test most of NaniBD features. We distinguish to different phases: pre-processing, which is performed with ProcBD prior to the revision procedure; and the revision procedure itself which is carried out with RevBD.

Pre-Processing Phase with ProcBD

The pre-processing of the Catalonian corpus uses just one command per speaker. This command calls successively other scripts that perform the following actions:

- Speech-silence detection: in order to reduce the length of the signal files listened to by no considering the silent parts at the extremes (this can be overridden at revision time).
- Signal to noise ratio and clipping factor determination for every file of the speaker. This information is shown and used later during the revision.
- Automatic recognition of the identification number of the speaker: this number is needed in order to know which are the theoretical orthographic contents of his files.
- Fetching of the orthographic contents, using the identification number automatically recognised. If the recognition process failed, the reviser is still able to fetch the correct ones at revision time.
- Automatic recognition of other spontaneous answers (yes/no, telephone number and identity card number).

This procedure is scheduled to be done each night with all the speakers and files pending. It has been proved to be quite effective, mostly due to the silence detection and yes/no answers.

Revision Phase with RevBD

During the revision phase of the Catalonian corpus, the reviser is told to annotate not only the orthographic contents of the files, but the speaker's data as well. The data asked are all four of the, i.e. the speaker's sex, year of birth and dialect, and the recording environment. The files used to configure these data are shown on Figures 3-5.

In order to facilitate both the annotation of the othographic text and the speaker's data entering, several extern commands are scheduled at different times during the revision.

```
Masculi
Femeni
Ambigu
```

```
Ambigu
Domicili
Public
Cabina
Mobil
```

```
Central
Nordoccidental
Balear
Valencia
Desconegut
```

Figure 3-5: Sex, Environment and Dialect files used in Catalonian corpus

- Theoretical contents fetching. In the Catalonian corpus, the identification number is needed to fetch the theoretical contents of the files. In order to deal with changes in the Identification Number during the revision, we use CmdNumId which automatically the fetches the correct files whenever the reviser makes a changes.
- Arabic numbers and uppercase letters conversion to text. By calling the external command under demand (option –D CmdSenDem, of RevBD) the reviser only has to press Ctrl-J to perform automatic numbers and letter translation.
- Determination of the speaker's dialect. In the Catalonian corpus, there is a quite straight relation between the dialect of the speaker and one of his spontaneous answers: the city where he spent most of his infancy. Nevertheless, this information is, sometimes, difficult to deal with. In our case, many answers do not respond to the question done, and sometimes there are ambiguities in the orthography of the city names. Our CmdSenDem, when called with the file that includes this answer, reads the contents of the orthographic transcription, and performs a search on a Catalonian city database. It allows the same syntax as UNIX command /bin/egrep. If an exact match is found in the database, nothing is output in screen, so as the execution doesn't lock, and the file with the dialect of the speaker is overrode with the corresponding dialect following the database criterion. If there is no match or it is not exact (it contains wildchars), the matches are shown at screen and execution locks until the reviser enters a key. In this way, the reviser may use this database search engine not only in order to know the dialect but to know the correct spelling of the name as well.

Acknowledgements

This work has been supported in part by the grant TIC97-0271-CE of the Spanish government and by another one of the CREL (Centre De *Referència en Enginyeria Lingüística)* of the Catalonian government.

The authors wish to thank Prof. Climent Nadeu for his valuable suggestions.