# Annotation of a multichannel noisy speech corpus

## L. Cristoforetti, M. Matassoni, M. Omologo P. Svaizer, E. Zovato

ITC-irst (Istituto per la Ricerca Scientifica e Tecnologica) Povo I-38050 Trento, Italy {cristofo,matasso,omologo,svaizer,zovato}@itc.it

#### Abstract

This paper describes the activity of annotation of an Italian corpus of in-car speech material, with specific reference to the JavaSgram tool, developed with the purpose of annotating multichannel speech corpora. Some pre/post processing tools used with JavaSgram are briefly described together with a synthetic description of the annotation criteria which were adopted. The final objective is that of using the resulting corpus for training and testing a hands-free speech recognizer under development.

## 1. Introduction

This paper deals with development and use of tools for annotation of a multichannel speech corpus collected in car environment under the European projects VODIS-II and SpeechDatCar.

This corpus is characterized by a five channel recording of each utterance (a close-talk microphone, three in-car microphones, and a microphone connected to GSM). As a consequence, it was important to have an annotation tool that allows visualizing and transcribing at the same time all the multichannel signals.

A new tool, called **JavaSgram**, was developed with the objective of being: flexible for independent annotation of different channels, fast in visualizing and playing input signals, fast in providing the transcriber with all the information needed for rapid successive annotation of all the items belonging to a given recording session. The tool was written in Java in order to ensure easy portability to different platforms and operating systems. To further speed-up the annotation task, JavaSgram was then enriched with some automatic pre/post-processing tools, as automatic channel alignment and utterance segmentation. So far, JavaSgram has been successfully used under Linux and Windows 95/NT platforms, both at IRST laboratories and at other labs involved in the SpeechDatCar project.

In the following, a brief description of the annotation activity under way will also be given.

## 2. VODIS-II/SpeechDatCar Italian corpus

The VODIS-II/SpeechDatCar Italian corpus consists of 600 sessions (300 speakers x 2 sessions) to be recorded under various noisy car conditions (e.g. town-traffic, air-conditioning, etc.); till now, 380 sessions have been recorded. Each session consists of 125 items, including isolated words, spelled words, continuous speech sentences, spontaneous speech sentences, etc. Recordings are accomplished by using a multichannel acquisition board (close-talk + three far microphones) operating at 16 kHz sampling frequency/16 bit accuracy per channel; at the same time a far-microphone, connected to a specific equipment, allows remote recording of a GSM version (at 8 kHz s.f./8bit A-law compression). Clearly, this signal is not synchronized with the in-car ones.

A more detailed description of the whole corpus can be found in (H. van den Heuvel et al.).

## 3. JavaSgram annotation tool

JavaSgram is a tool, developed at IRST with the objective of being flexible for independent annotation of multiple input channels. It is a generic annotation tool in which several general features are combined with dedicated features to allow a fast and effective annotation of the VODIS-II/SpeechDatCar database.

JavaSgram was developed entirely in Java, using the Sun JDK 1.6 under a Linux platform. This allows a portability through several operating systems without changing the code. Performance was improved with the use of Just In Time compilers and is now adequate to allow a effective use of the tool without strong power requirements.

#### 3.1. General features

Javasgram is provided with some features developed to handle generic data files, with single or multiplexed channels; these features are described in the following.

The tool has the capability to handle different file formats: raw PCM, NIST-SPHERE, SpeechDatCar file format. Each sample is represented by 16 bit (signed or unsigned, Big Endian or Little Endian byte order); sampling frequency ranges from 8000 to 44100 Hz.

The application can simultaneously open and display up to eight signals, being these signals either multiplexed in a single file or stored in different files with a determined naming rule. As an example, Figure 1 shows the use of the tool with a single channel file.



Figure 1: The JavaSgram tool opening a single data file.

The **Play** section allows to listen to the various channels in an independent way. It is possible to play the whole signal or just a part of it. The selected part can be delimited by markers or selected by mouse. It is also possible to play all the segments delimited by markers, with a little delay between successive segments.

Several **Zoom** capabilities are provided with the tool. The signals are always zoomed together in parallel. It is possible to select the area to zoom using the mouse or entering the correct interval with a dialog window. A feature then allows to slide the zoomed area to the left or the right of the current part of the displayed signal. This is an easy way to analyze the whole signal looking just at a small piece of it at a time. In the Zoom menu two other useful options are provided, to measure the duration (both in samples and in milliseconds) of a segment either delimited by two markers or selected by the mouse.

As speech segmentation may be useful for training speech recognizers, JavaSgram was conceived to allow insertion of markers either in synchronous or in asynchronous mode, in this way making evident events occurring in all of the channels as well as in a single channel. In case of cardata, examples of the latter situation are:

- a burst event caused by the road conditions, not audible in the close-talk microphone signal.
- a lip noise caused by the speaker, audible only in the close-talk microphone signal.
- a distortion due to the GSM transmission, audible only in the GSM microphone signal.

The segmentation through markers is saved in different files for different channels, in order to ensure the full independence of the data. These files area automatically loaded when a signal is opened. An external program provides an automatic segmentation of the signal and the user can act on the proposed segmentation modifying it as needed.

Related to the segmentation there is the labelling section (**Label** menu). A specific transcription (consisting of text and/or symbols) can be assigned to every segment of the given signal, a segment being delimited by two markers. Transcription can be edited directly via a dialog window or can be chosen from an external file, separately loaded. The use of this file reduces the human typing errors and ensures coherent writing rules.

#### **3.2. Dedicated features**

JavaSgram tool was then enriched with some features, especially developed to speed up the annotation process of the given database.

During recordings, the four in-car signals are multiplexed in a single file while the A-law GSM telephone signal is stored in a separate file. Then, these files are put together in a single five-channel file, converting the GSM signal to the same format of the other signals and realigning them. The complete procedure is described in section 4.1. The resulting file can be opened once the options of the tool are set to handle the correct data format.

Figure 2 shows an example of use with a multichannel file of VODIS-II/SpeechDatCar database. The first signal is referred to the close-talk microphone, then there are three far-microphone signals, and the last one is the GSM signal. It is possible to obtain the alignment of the signals, by inserting padding zero samples at the head and tail of the incar signals, according to the cross-channel delay measured previously, as discussed in section 4.1. It is worth noting the presence, in all the channels, of DTMF tones, necessary for synchronizing fixed and mobile platforms during recordings, but which would not be sufficient to derive an adequate synchronization for other purposes as annotation. Figure 2 also shows the two markers delimiting the speech segment, provisionally identified with < ITEM > and that will be successively replaced with the correct transcription provided by the annotator.



Figure 2: JavaSgram opening a 5-channel multiplexed file.

An important feature is the integration of the data files with the documentation files (SAM files). When an interleaved data file is opened, the corresponding SAM files for the in-car signals and the GSM signal are opened at the same time. Once the annotation is complete, the SAM files are updated with the information on segmentation and labelling. The SAM files are used also to display information about the recording conditions, which can be useful during annotation to understand some spurious phenomena.

The most important feature introduced is the labelling window (see Figure 3). The text that should have been read by the speaker is contained in a field of the relative SAM file. This text (together with other possible symbols) is extracted and showed to the annotator, who can decide to validate the resulting transcription or to modify it according to what was really uttered by the speaker. This is very timeefficient, because if the speaker read correctly the prompt and no spurious phenomena were present in the signals, just a mouse click is needed to update the SAM files and to load a new file on the window. On the other hand, if a phenomenon is audible only in one channel, it is possible to edit the related transcription, by entering another window that shows text and symbols of each channel. Figure 4 shows the window in which the annotator edits the tran-

| Propased Label            | x                           |
|---------------------------|-----------------------------|
| [dit] <urgente></urgente> | e label and press Ok        |
| Play Play Se              | g Check Ok Cancel           |
| File Info                 |                             |
| Speech File Manue         | V1145120JTV                 |
| Corpus Code               | 20                          |
| Speaker Code              | 149                         |
| Scenario Code             | TOWN, TRAFFIC               |
| Microphone Type           | CHNO-SHURE, CHNI-AKG, CHN2- |

Figure 3: Primary labelling window.

scription for each channel. In this example the close-talk transcription has been modified with the insertion of a specific symbol [spk], used to categorize sounds produced by the speaker. The labels relative to the far-microphones present a symbol that states the presence of an intermittent noise caused by the road conditions (i.e. in case of a rough road). This kind of noise is labelled with the symbol [int]. The GSM transcription instead includes several [dit] symbols, which identify DTMF tones.

To speed up the labelling process, a mask containing a pattern for each channel was introduced in the JavaSgram tool. That mask can include text/symbols which are automatically proposed to the annotator. For example, some synchronization DTMF tone sequences are present only in the GSM channel. They can be written in the mask, in such a way they are put automatically in the GSM transcription. This is useful even to account for stationary phenomena that can be present in the entire session. As an example a rain noise symbol, likely to be used for all of the recordings of a given session, may be inserted in the mask at the beginning of the annotation session.

To make the annotation less prone to human errors, other automatic pre-processing features are available, for istance to convert automatically digit strings in their corresponding orthographic representation. This conversion has been implemented for three languages: English, Italian and French. This option avoids the necessity of manually transcribing digit sequences in words.

Another useful feature is the possible correction of transcriptions of signals already annotated. In practice, it is possible to choose if SAM files must be read as original ones (those produced by the platforms during recordings) or as annotated ones. In the latter case, transcriptions are presented to the annotator, who can modify them and save the re-edited SAM file. For what concerns segmentation, the annotator has also the possibility to adjust the given markers if they are not accurately placed.

## 4. Pre-processing tools

#### 4.1. Automatic aligning and multiplexing

In order to visualize all the in-car signals plus the GSM one (5 channels), a tool for aligning and multiplexing signals has been provided. First of all the GSM signal is up-

| ClaseTalk label: | [dt] [spk] «guettro . cinque . deo . entto . sei . novo . una . otto . zero . teo-                   |
|------------------|--|
| Civil label:     | [dt]-dir( quattra . cirque . due . sette . sei . neve . uno . etta . zero . tres                     |
| Chil label:      | [dit] djint] quattra . cinque . due . sette . sei . nove . uno . etta . pero . tre>                  |
| Cit/3 Moleit     | [dit] <[int] quattra . cirque . due . sette . sei . nove . uno . atta . zoro . trex                  |
| GSM label:       | [dit] [dit] -quettro , cinque , dao , setto , sei , novo , una , otto , zero , trov (dit] (dit] (dit |

Figure 4: Independent labelling window.

sampled to 16kHz (i.e. the sampling rate of all the other channels), and then converted to linear dynamics at 16 bit accuracy. Once the signal has been upsampled it is aligned to all the other channels.

For this purpose, a technique was used, based on the CrossPowerSpectrum Phase analysis (M. Omologo et al.), which yields the delay between two acoustic signals for which the phase correlation is highest. In this case the GSM and the close-talk channels have been used. Once the alignment has been completed the five signals are multiplexed in a unique 5-channel file. This alignment module is accurate, as it provides a correct result up to sample level.

### 4.2. Automatic segmentation

One of the related tool of JavaSgram is the automatic segmentation that provides a bounding of a speech utterance, of any length and content. In this way, the annotator has only to check if the suggested segmentation is correct and in case of error he has the chance to move the markers to the correct positions. The performance of this tool is quite good (97% of correct segmentation) and it is important to emphasize the fact that a good and consistent segmentation is useful for recognition purposes.

To achieve this result, three start end point detection algorithms have been combined; their outputs are compared one each other in order to make the best choice of start and end markers for the given signal.

## SVF-based algorithm

The first algorithm is based on the calculation of the Spectral Variation Function (F. Brugnara et al.), that is a measure of the distance between sequences of spectral coefficient vectors, which represents a speech segment of specific length. This choice is due to the fact that in the speech frames there is more spectral variability than in background noise frames. The spectral coefficients are calculated from the outputs of a filter bank. Each filter has a triangular bandpass frequency response and the bandwidth and spacing between central frequencies are determined by a constant interval in a mel scale. The frequency analysis is also used to detect (and skip) the DTMF tones present in the SpeechDatCar data.

#### **CSP-based algorithm**

The second algorithm is based on the CrosspowerSpectrum Phase analysis (M. Omologo et al.); in particular, the close-talk channel and one far microphone channel are used to measure a generalized cross-correlation between the two corresponding versions of the given pronounced item. In the background noise frames the correlation between channels is very low, on the contrary it grows significantly in the speech frames.

#### **Energy-based algorithm**

A third, energy based, algorithm has also been taken into account. Due to the low signal to noise energy ratio, it does not provide reliable boundaries in the case of car-data.

All these algorithms produce a first set of "islands" of probable speech segment. A subsequent specific procedure estimates the two most likely boundaries. For this purpose, the largest island is initially chosen as a first estimate. Then adjacent islands are considered, and if their distance from the main estimate does not exceed a given threshold, a new estimate is calculated. At the end, each algorithm provides the start and end marker hypotheses. The SVF output is taken as a reference and then the other outputs are compared. If a new marker is found that is external and not very far from the reference, this becomes the new reference. In order to optimize the performance of this tool, for any kind of signals, many parameters can be tuned, such as the maximum distance between islands of speech frames, the minimum number of frames for which an island is considered "speech", etc.

## 5. Annotation criteria

During annotation, some criteria were defined in order to ensure a coherent result from different transcribers. These criteria regard the confirmation of proposed automatic alignment and segmentation as well as the use of specific labels to indicate acoustic events (as noise caused by rain, wind, cobble stones, etc) which can be critical in carnoise speech recognition.

A description of the whole set of symbols adopted in SpeechDatCar project and being used to describe the various acoustic events goes beyond the purpose of this paper. Here we will emphasize the most common "labelling" situations examined by the annotators while working with the available portion of the VODIS-II/SpeechDatCar corpus. Some of the criteria described in the following refer to recommendations also defined by the SpeechDatCar consortium.

#### Stationary background noise

Recordings were accomplished under different noisy conditions. As a consequence, different background noise levels may be present in the various sessions. When the background noise is stationary and coherent with the recording conditions (indicated by a label in the SAM file) it is not further indicated in the annotation.

#### Non-stationary acoustic events

Several non-stationary events may be present in the recordings, among which those produced by the speaker (e.g. lip smack, cough, etc) or those produced, for instance, by the direction indicators, horn sounds, wheels running on rough road. For most of those events a specific symbol is used (e.g. [int] in case of generic intermittent noise). However, some of them may be extended along the recording, others may be concentrated in a short segment; further,

some of them may be concurrent with the speech message, others may occur before or after it. With the purpose of making the annotation more detailed, we have introduced some symbols which indicate specific non-stationary events and some symbols which give a better idea of the event time extension.

#### Time location of noise events

Nevertheless, the given symbols do not provide information of the time instants when the event begins and ends. For this purpose, a parallel annotation would be necessary. In our work we adopted a compromise solution, by introducing a symbol ordering which provides a better idea of the sequentiality of noise events and speech segments.

## 6. Conclusions

At this moment, 190 speakers (out of the 300 foreseen at the end of the project) have been collected and annotated. So far a portion of this material has been used for handsfree speech recognition training and testing purposes with successful results. Results concerning the use of the first 25 speakers are reported in (M. Matassoni et al.) and show that the use of this material is of fundamental importance for the development of an in-car speech recognition technology. The above referenced results also show that the annotation accuracy as well as the annotation criteria adopted at our labs ensure additional benefits to those which would be intrinsically provided by the speech material itself.

## 7. Grants and funding

Part of SpeechDatCar and Vodis-II projects is funded by the Commission of the EC, Telematics Applications Programme, Language Engineering, Contracts LE4-8334 and LE4-8336. The JavaSgram tool is distributed by means of a license to ask to ITC-irst.

### 8. References

- H. van den Heuvel et al. "SpeechDat-Car: towards a collection of speech databases for automotive environments", *Proc. of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, Tampere (Finland), May 1999, pp. 135-138.
- F. Brugnara and D. Falavigna and M. Omologo "Automatic Segmentation and Labeling of Speech Based on Hidden Markov Models", Speech Communication, Vol. 12, pp. 357–370, 1993.
- M. Omologo, P. Svaizer, "Use of the Crosspower-Spectrum Phase in Acoustic Event Location", *IEEE Trans. on Speech and Audio Processing*, May 1997, vol. 5, n. 3, pp. 288-292.
- M. Matassoni, M. Omologo, L. Cristoforetti, D. Giuliani, P. Svaizer, E. Trentin, E. Zovato "Some results on the development of a hands-free speech recognizer for carenvironment" In *Proc. of Automatic Speech Recognition* and Understanding Workshop, December 1999, Keystone (Colorado).