Transcriber: development and use of a tool for assisting speech corpora production

Claude Barras[1], Edouard Geoffrois[1], Zhibiao Wu[2]

[1]DGA/CTA/GIP, 16 bis av. Prieur de la Côte d'Or, 94114 Arcueil cedex, FRANCE

[2]LDC, 3615 Market Street, Suite 200, Philadelphia, PA, 19104-2608, USA.

Abstract

We present the development and use of "Transcriber", a tool for assisting the creation of speech corpora. It is more specifically geared toward the manual segmentation and transcription of long duration broadcast news recordings, including annotation of speech turns, topics and acoustic conditions. It is distributed as free software in order to encourage the production of corpora, ease their sharing, increase user feedback and motivate software contributions. It is highly portable, relying on the scripting language Tcl/Tk with extensions such as Snack for advanced audio functions and tcLex for lexical analysis, and it has been tested on various Unix systems and Windows. The data format follows the XML standard, and Unicode is supported for multilingual transcriptions. The use of a scripting language allowed a rapid prototyping development mode, with regular feedback from users and quick adaptation to their needs. Transcriber has been used for over a year, and we present the experience gained. There are now many users in several countries, and new needs appear. Future developments include the generalization of the data format and of the tool using the recent reflections about annotation graphs which is expected to

further increase flexibility.

1 Introduction

Speech research has long been conducted using small- or medium-sized databases recorded in controlled conditions. The transcription of these databases was not a critical issue, since the content was known in advance or utterances were short enough to be easily transcribed. With the advent of work on realistic speech — spontaneous speech with frequent overlaps and uncontrolled background conditions —, this situation has changed. An always larger amount of transcribed speech is required for the improvement of automatic speech recognition systems and their adaptation to new tasks. For example, several hundred hours of manually transcribed speech have been used for the Broadcast News task in English within the DARPA programs. Production of these data requires a suitable software environment.

A new project for transcription and indexation of multilingual Broadcast News started at DGA in 1997, and such a software environment was needed for creating the necessary corpora. After examination of existing solutions, it appeared that no available transcription software completely filled our needs, and we decided to develop a new tool. The development of "Transcriber" started at the DGA in coordination with the LDC in early 1998. This tool allows to manually segment, label and transcribe long duration speech signals, including labeling of speech turns and topic changes. It is multi-platform and it is distributed as free software [16]. The first release was presented at the LREC conference in 1998 [1]. Since then a lot of evolutions have taken place, and we present the current status of the tool and the experience gained.

In section 2, we present the requirements identified for the tool. In section 3, we present its development, especially the choice of a programming language,

and the solutions chosen for an efficient multi-platform audio management and for the data format. The main features of the tool and its interface are described in section 4. Our experience of using the tool is presented in section 5. Future directions and format evolution are discussed in section 6. We conclude with a discussion on the benefits we found in this experience.

2 Requirements

We identified the following technical requirements for the tool: it should provide interactive management of long duration signals, allow multilingual transcriptions, be user-friendly and usable by a non-specialist, and work on a standard PC. Furthermore, a large diffusion of the tool appeared beneficial for the project, and it was decided that the tool should be freely distributed, be portable and use a standard format.

2.1 Interactive management of long signals

Interactivity of a software tool is generally considered as a practical issue being less important than interface design, but quick responses to user actions is a condition for the tool to be accepted by the user [9]. In the case of Broadcast News projects, typical recording duration extend from several minutes to even hours. An interactive access to such long signals with user-reactive playback, scrolling and zoom should be available. In order to help the segmentation process, showing a moving cursor perfectly synchronized on the signal waveform during playback also seems necessary; implementation of this feature is often platform-specific.

2.2 Multilingual transcriptions

Our project of multilingual indexation implies management of several languages in the tool and also in the data produced, including non European languages. Obviously, Unicode which is the most standard multilingual character encoding should be supported.

2.3 User-friendly interface

Transcribing audio or video recordings is a very time-consuming task which can not be left to speech scientists. It is usually done by educated, native people of the studied language with no specific skill in computers science. A transcription tool should mimic as much as possible user interfaces of standard office software, so as to reduce the training period. It has to be intuitive, and a good integration between sound control and the annotations must be provided.

2.4 Working on standard computers

Another wish was that it should work on a low-cost, standard computer, in order to reduce the additional cost per workplace. Previous solutions for Broadcast News were designed to work only on higher-end workstations, but this led to a lack of portability. Furthermore, the transcribers could only complete their work in the office. Using a standard PC (desktop or even laptop) is more economical and makes work at home possible.

2.5 Free distribution

Apart from the technical or economical constraints mentioned above, we chose to distribute the tool as free software under the GNU public license. Indeed, having decided to develop a new tool, the additional cost for distributing it is modest and important benefits can be expected in return. The main benefit is

to encourage the production of speech corpora and ease their sharing. Another reason is the efficiency of open source for software development. We expected an increase of user feedback and also some contributions by external developers.

2.6 Portability

The portability of the tool became a more important issue with its diffusion as free software. Like many research laboratories we are mainly working under the Unix operating system, and we chose to make the development under Linux, which is a free, widespread Unix for PC. However, Windows is by far the most common system outside laboratories, and having the tool run also under Windows was very soon felt as important, since most transcribers are generally accustomed with this environment and do not wish to switch to another system. A multi-platform tool is needed. Even if the drawbacks can be that some compromises have to be done on the efficiency of the software or that the maintenance cost for several platforms gets higher, designing portable code is always a good practice, especially in the long term.

2.7 Use of standard formats

The choice of a good format for the annotations is also important, if only for their diffusion; it should as much as possible follow existing standards. Since there is a large number of existing formats for the annotations, this choice is not obvious.

3 Development

This section presents the main development choices which were made, in line with the above requirements.

3.1 Choice of Tcl/Tk scripting language

We were confronted with the choice of a language for the development. Visual Basic and Visual C++ from Microsoft are widely used in the industry for the development of applications under Windows, but this choice limits the portability to other systems. Graphical applications under Unix classically involves C language with X11 interface, but the development of user interfaces in this framework is generally complex, and the programs are also not easily portable to Windows. Those two solutions were not really considered.

Java was a possibility, since it is a modern, multi-platform language. An annotation tool developed at LDC was indeed implemented in Java a few years ago. However, compared with softwares such as Xwaves, the waveform display and its update according to user requests were slow; it could not simultaneously display a moving cursor during playback at the time of development, and first version of Java could only support AU sound file format. This is why the LDC finally dropped the idea of implementing a fully functional package in Java. Furthermore, the status and the licensing policy of Java and of some libraries needed for the user interface or audio management were unclear at the time of choosing a language for a new development.

Since a few years ago, there has also been a growing interest for various scripting languages [11]. One of the most open and successful ones is Tcl/Tk. It is a multi-platform script language available for several Unix systems, Macintosh and Windows [10][14]. The syntax of the Tcl language is rather simple, but complex user interface can be written in a few lines using the Tk graphical library. The absence of compilation significantly speeds up the development process, and computers have become so powerful that interpreted applications provide nowadays very reactive user interfaces. The need for a C or C++ development is reduced to the critical or system-dependent parts which can

easily be interfaced with the Tcl script. Tcl/Tk was therefore chosen for the development of Transcriber. In its version 1.4, Transcriber accounts for about 12.000 lines in Tcl and about 3.000 lines in C, which means that most parts were responsive enough in Tcl.

3.2 Interactive display of long duration waveforms

As previously stated, it was necessary to provide interactive display and playback of long duration signals. Scrolling and if possible zooming of the waveform had to be achieved in real time on a standard platform.

A specific waveform display module has been developed for Transcriber. This part is time critical and is written in the C language. It is optimized for interactive zooming and scrolling on long duration sound files, which can be performed even while listening. The sound file is never loaded in memory, since a single hour of signal could easily exceed the available memory. The first time a long sound file is accessed, the temporal envelope of the waveform is computed at a low resolution (minimal and maximal sample values for each 10 ms segment) and stored on the disk in order to speed-up later display. When drawing the waveform at low resolutions (i.e. several minutes of signal or more), its shape is computed using only the pre-computed envelope instead of reading megabytes of data in the sound file. Also, only the needed part of the waveform is computed during scrolling, not the whole display. These optimizations proved to increase dramatically the interactivity of zooming and scrolling.

As an option, remote sound file access is provided through a server controlled with sockets and is specifically optimized for the tool, thus being more efficient than a standard network file access: for signal display, the waveform is computed on the server and is transmitted over the network instead of accessing the whole signal through the network. In our experience, the annotation tool was

always used in a stand-alone fashion. However, this feature makes it possible to centralize all recordings on a server, allowing interactive remote access without duplication of large resources. In a predictable future, it will probably become more useful for the consultation of archives.

3.3 Audio management with Snack

A cursor should also be synchronized with playback. But low-level access to the audio driver is generally needed in order to obtain the latter behavior, and makes portability harder. Much time was spent during development and testings before having a reliable sound control, especially because of hardware or of low-level OS problems (e.g. under Linux, early versions of the audio driver were not completely safe and frequent reboot were needed before identifying the problem, or some recent cards were not correctly recognized). The Snack audio extension brought a good solution to the multi-platform audio question.

Snack is an extension for the Tcl/Tk scripting language which provides multi-platform audio management [12]. It was developed by K. Sjölander at KTH speech laboratory and its use has been illustrated for educational purpose [13]. Most commonly used sound file formats are supported, playback is efficiently supported for Windows and several Unix systems including Linux, and it runs in the background while staying under the control of the application. These excellent technical characteristics and the fact that it is distributed as free software made Snack obviously the best choice for multi-platform audio management. It was thus chosen for use within Transcriber as soon as it was available, and most of the code initially developed was dropped.

The possibility of remote playback was developed in the first version of the tool in order to allow using it from a terminal [1]; this feature didn't prove to be a big need, and was not maintained in later versions of the tool.

3.4 XML file format

The produced transcription consists in a set of annotations which apply to the audio recording. These data need to be stored in a file, processed in various ways, and exchanged easily. The data format has thus to be chosen carefully, since frequent changes would be inconvenient for the users. The kind of data to be stored must of course be considered before defining the format. Our primary task being automatic transcription of broadcast news, we were mainly interested in an orthographic transcription along with a description of various acoustic events, speech turns with an identification of the speaker, and a broad segmentation in topics.

The choice of a human-readable format was easy, since textual data prevail in our data, and anyway binary format are always difficult to share. But transcriptions are complex objects, and a structured machine-readable format is also needed. We considered SGML and its more recent subset XML [7]. Both allow to describe a document structured as a tree. Each node of the tree bears a set of attributes with a value. The syntax used in the document can be specified in a Document Type Declaration or DTD and distributed along with the document. Tools exist for ensuring automatically the well-formedness and validity of a document, i.e. the fact that it correctly follows the SGML or XML syntax as well as its specific DTD. More important, SGML and XML are very widespread standards, which helps sharing documents in this format. XML uses Unicode character codes, which is very important for multilingual transcriptions. Compared to SGML, automatic processing of XML documents is much easier, and XML was thus preferred.

However, XML normalizes the way data are stored in a document, not which data are stored nor what they mean. The design of a DTD is the preferred way to constrain a document to a specific syntax or structure. The way this structure

has to be created, displayed, managed, is still left to a specific application or to the user.

3.5 Implementation of the parser

Then we looked for an XML parser being:

- easy to interface with Tcl/Tk,
- in open source for a diffusion along with our tool,
- validating the document according to its DTD, since the production of valid documents is needed for their automatic exploitation.

At the time of development, no completely free validating XML parser was available for Tcl/Tk. A specific one was therefore designed, as a separate module with the following features:

- it makes use of tcLex, a lexical analyzer generator extension to Tcl and distributed as free software [5]. The tcLex extension is inspired by Unix and GNU lex and flex, but follows Tcl syntax and suppresses the need for compilation of the analyzer; this feature proved to be very convenient for the development of the XML parser.
- an image of the read XML file is stored in memory in a tree data structure and in an object-oriented fashion, each node of the tree being accessed to and modified by methods.
- when a DTD is active, each modification of the XML data structure in memory is validated immediately; this ensures that saving the current XML image to a file will produce a valid XML file.

Furthermore, the choice was made to always keep the XML data structure in memory up-to-date with the current state of the transcription. The advantage is

that saving the transcription only requires a dump of the existing data, which in addition is already ensured to be valid; some parts of the document (comments, tags intended for another use) can also be ignored by the application and still remain in the document. The drawback is that the XML object-oriented tree structure used is not always the most convenient way for data management in Tcl (which leads to some duplication of the data in memory), and that the tool is much more sensitive to the choice of the DTD and its possible modifications. The relevance of this choice remains hard to evaluate, and would depends highly of the internal structure for other applications.

With the growing interest around XML and the trend of normalization in the area of programming interface to manipulate XML documents (e.g. with the Document Object Model or DOM [6]), using another existing XML parser would have to be considered. Indeed, relying on robust libraries tested and used by a lot of users is one of the main advantages of open-source developments. XML parsing and data management presently accounts for about 1/5th of the total Tcl code in Transcriber, and using an external module would reduce the maintenance workload for this part.

4 Main features

Combined with the free distribution, the use of a scripting language allowed rapid prototyping development with quick user feedback on the tool. Numerous functions were modified or added according to user requests. For example, management of overlapping speech was changed several times in order to provide a more intuitive user interface. This development mode lasted over a year with monthly updates. The resulting tool is now briefly described, with more details on the features relevant to the structure of speech annotation

4.1 User interface

Only a brief description of the user interface and of the main features is given here. A more complete documentation is distributed along with the tool.

The user interface of the tool consists mainly in two parts (cf. Figure 1):

- a text editor in the upper half of the screen, for creation, display, and edition of the transcription;
- a signal viewer in the lower half of the screen, along with the temporal segmentation at different levels (orthographic transcription, speech turn, topic change, background noise).

Playback is controlled by tape-recorder-like buttons and by keyboard shortcuts. Any portion of the signal can be selected, played, or zoomed. A cursor is synchronized with current position during playback.

The segmentation of the signal is done using a menu or simple keyboard shortcuts and is even possible during playback without stopping it; this allows a quick rough segmentation in a first phase, and a more precise position of segment boundaries in a second phase by simply dragging the boundaries under the signal with the mouse.

Each segment boundary defined in the signal editor appears as a graphical mark in the text editor, and the text typed in between two marks belongs to the segment. Some boundaries can be defined as the beginning of a new speech turn or of a new topic; a button appears in the text editor above the time mark and can be clicked on for further editing.

Any change in the text editor is immediately displayed under the signal in the temporal segmentation. Also, cursor position in the text editor and in the signal viewer are synchronized and are constrained to always stay within the same temporal segment: as soon as one cursor moves and switches to another

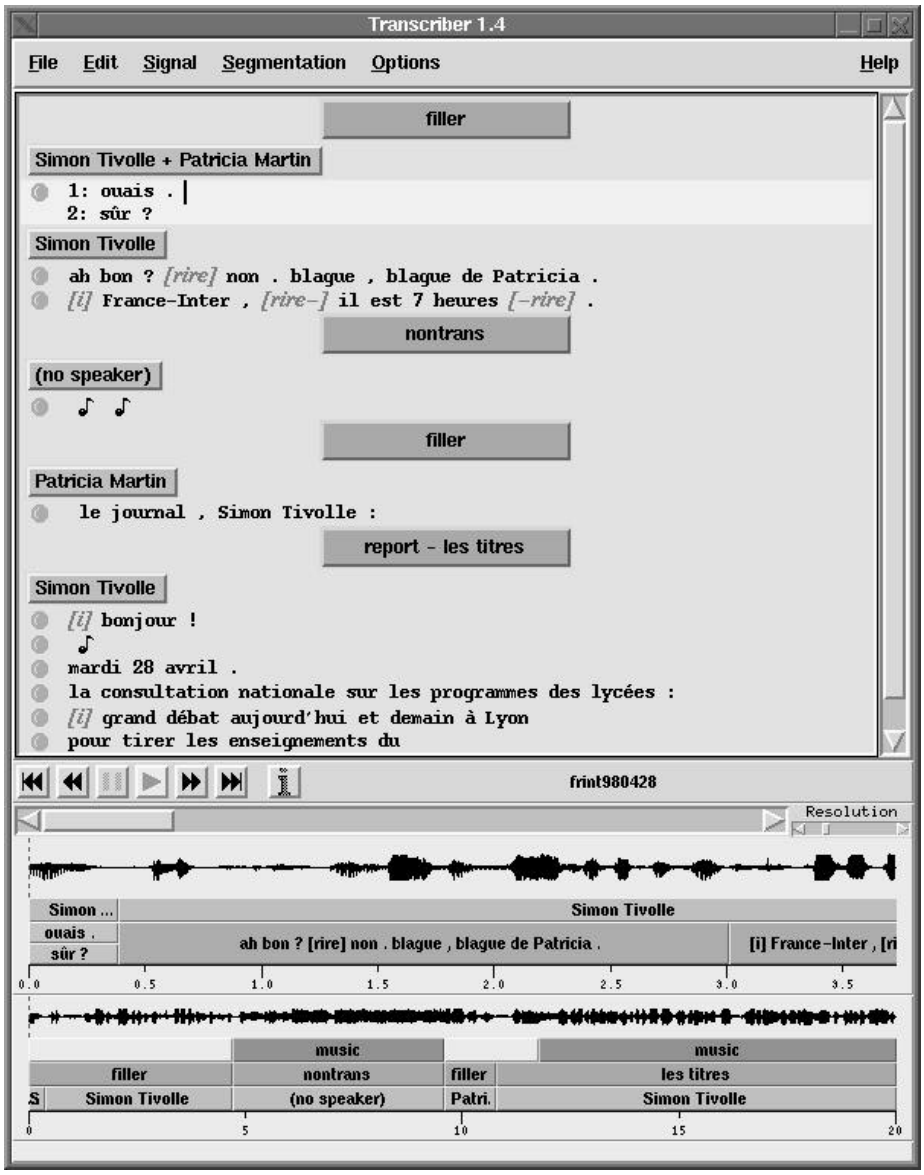

Figure 1: Screen shot of the user interface.

```
<?xml version="1.0"?>
<!DOCTYPE Trans SYSTEM "trans-13.dtd">
<Trans version="1" version_date="981211"
    audio_filename="frint980428" scribe="YM" xml:lang="fr">
  <Topics>
    <Topic id="to1" desc="les titres"/>                          ] List of
  </Topics>                                                      ] topics
  <Speakers>
    <Speaker id="sp1" name="Simon Tivolle" type="male"/>         ] List of
    <Speaker id="sp2" name="Patricia Martin" type="female"/>     ] speakers
  </Speakers>
  <Episode program="France Inter" air_date="980428:0700">
    <Section type="filler" startTime="0.000" endTime="4.736">
      <Turn speaker="sp1 sp2" startTime="0.000" endTime="0.387">
        <Sync time="0.000"/>
        <Who nb="1"/>
        ouais .
        <Who nb="2"/>
        sûr ?
      </Turn>
      <Turn speaker="sp1" startTime="0.387" endTime="4.736">
        <Sync time="0.387"/>
        ah bon ?
        <Event desc="rire"/>
        non . blague , blague de Patricia .
        <Sync time="3.008"/>
        <Event desc="i"/>
        France-Inter ,
        <Event desc="rire" type="noise" extent="begin"/>
        il est 7 heures
        <Event desc="rire" type="noise" extent="end"/>
         .
      </Turn>
    </Section>
    <Section type="nontrans" startTime="4.736" endTime="9.609">
      <Turn startTime="4.736" endTime="9.609">
        <Sync time="4.736"/>
        <Background time="4.736" type="music" level="high"/>
        <Background time="9.609" type="other" level="off"/>
      </Turn>
    </Section>
    <Section type="filler" startTime="9.609" endTime="10.790">
      <Turn speaker="sp2" startTime="9.609" endTime="10.790">
        <Sync time="9.609"/>
        le journal , Simon Tivolle :
      </Turn>
    </Section>
    <Section type="report" topic="to1" startTime="10.790" endTime="20.000">
      <Turn speaker="sp1" startTime="10.790" endTime="20.000">
        <Sync time="10.790"/>
        <Event desc="i"/>
         bonjour !
        <Sync time="11.781"/>
        <Background time="11.781" type="music" level="high"/>
        <Sync time="12.237"/>
        mardi 28 avril .
        <Sync time="13.344"/>
        la consultation nationale sur les programmes des lycées :
        <Sync time="16.236"/>
        <Event desc="i"/>
         grand débat aujourd'hui et demain à Lyon
        <Sync time="18.521"/>
        pour tirer les enseignements du ...
      </Turn>
    </Section>
  </Episode>
</Trans>                                                         ] Transcription
```

Figure 2: Sample of a transcription file.

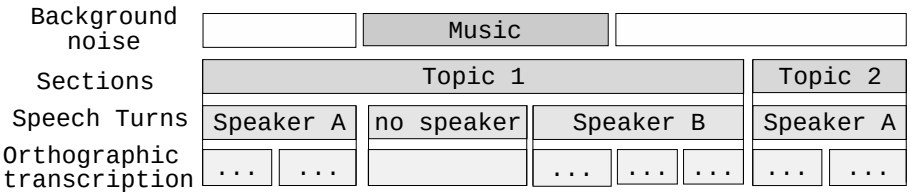

Figure 3: The 4 segmentation levels of a transcription.

segment, the other cursor automatically moves to the same segment.

4.2 Data format

We considered existing formats, especially the ones used at LDC for the DARPA evaluations on Broadcast News. In fact, a lot of other formats are currently in use. As an attempt to better coordinate existing efforts, the Text Encoding Initiative or TEI provided in 1994 recommendations for the transcription of written and also spoken materials in SGML [15]. TEI has not been followed in the design of the current Transcriber format, mainly because existing LDC formats were less complex and already adapted to the task, but it contains a lot of relevant ideas and propositions, and current efforts aim at adapting TEI to XML and expanding its coverage.

LDC has used SGML formats for several years. More recently, NIST specified an Universal Transcription Format or UTF[1] based on previous SGML formats used at LDC; it was designed for production of Hub-4 Broadcast News and Hub-5 Conversational speech corpora in 1998. After considering it, we chose not to use it directly for several reasons: we wanted to store some informations which are not available in UTF, e.g. some characteristics associated to the speakers; also, UTF uses SGML features which are not available in XML, e.g. short references or case-independence of identifiers. Nevertheless, we tried to keep a logical structure as near as possible to UTF or previous LDC formats and

[1] not to be confused with Unicode character encodings UTF-8 and UTF-16.

tag or attribute names are often reminiscent of LDC's ones, so that conversions between the different formats are straightforward (though partially lossy).

The transcription has three hierarchically embedded layers of segmentation:

- a basic segmentation of the orthographic transcription, with breakpoints at pauses, breaths, sentences or any other convenient places;
- a division into speaker turns;
- a division into larger sections, such as "stories" in the Broadcast News setting.

In addition, there is a fourth type of segmentation, hierarchically independent of the other three, for changing acoustic background conditions. Each of the four types of segmentation is constrained to be a partition of the whole signal (cf. Figure 3).

Speakers and topics are grouped in separate lists and each speaker or topic bears a unique identifier which is used inside the transcription; this avoids the duplication of speaker or topic names and makes them easier to edit.

We give a manually indented sample of a transcription file in the chosen format (cf. Figure 2); this is the one displayed in the previous screen shot of the tool (cf. Figure 1). In our case, the validation of a document is not enough to ensure its logical consistency; indeed, some properties — e.g. the fact that the "startTime" and "endTime" attributes must bear numerical values which are in increasing order — can't be specified in the DTD and have to be verified afterwards in the application.

4.3 Non-speech events and background noise

Main speaker's vocal non-speech events were initially typed in within the orthographic transcription as special markers between square brackets - e.g. `[i]`

for an inspiration. Such events normally do not overlap with speech, and they can be inserted rather naturally between the uttered words in the transcription. External short noises could also be noted in the same way, e.g. `[b]` for a generic noise. Such events can overlap with speech and they often do, so beginning and end of a noise were written in a simple way, e.g. `[b-] ... [-b]`; as an alternative for short noises overlapping with a single word, it could also be written joined to the word. This was also used for language changes. Keyboard shortcuts were predefined for some frequent events.

This mechanism was easy to use by the transcriber but had several drawbacks. Different kinds of events were noted the same way and could not be automatically distinguished, too much variability appeared in the event description, and further processing was needed in order to separate these annotations from the orthographic transcription. At some point it was decided to provide a specific interface within the tool for events management, and to distinguish the internal representation of the non-speech events or of various local annotations from their interface in the tool. As much as possible the initial display and keyboard shortcuts were kept identical, but the data format was extended with a new tag: several kinds of annotations were distinguished - noise, language change, lexical annotation, pronunciation annotation, comment, and a field was added to specify the temporal extent of the annotation. A set of predefined descriptions for each kind of event was made available to the transcriber, but any other description can be given.

The event description remains specific to the task and to the transcriber's language. In the future, it could be possible to agree on an international set of non-speech events or other annotations, with a localized display within the tool. This would ease the international exchange of produced corpora. But deciding which annotations are language-independent is not straightforward. At any

rate, the transcriber should remain able to add his or her own annotations. Also, the transcriber's workload has to be considered and mouse or keyboard manipulations should be kept minimal.

A different mechanism allows for a description of long duration background noise. A specific level of segmentation is displayed under the signal and music icon appears in the text inside the transcription where the background acoustic conditions change.

4.4 Overlapping speech

A mechanism is provided for the annotation and transcription of overlapping speech. Our priority was the transcription of single-channel Broadcast News recordings for speech recognition systems training, and within this framework overlapping speech segments are currently discarded from further automatic exploitation. However, future tasks may use them, it makes the transcription more complete, and anyway it was judged less frustrating for the transcriber to be able to transcribe overlapping speech, whether he uses this feature or not.

It proved to be difficult to provide an ergonomic user interface in the tool for this task. In a first implementation, the constraint imposed to the segmentations to be a strict partition of the signal was relaxed, and the last speech segment of one turn could overlap with the first speech segment of the next turn (cf. solution 1 in Figure 4). The overlapping segments could be drawn in the temporal segmentation under the signal, but the display of this solution in the text editor was confusing, because the two overlapping speech segments belonged to two separate speech turns and their simultaneousness did not appear clearly enough. Several interfaces were tried and changed at user's request before eventually choosing another representation (cf. solution 2 in Figure 4). The overlapping part is clearly marked as a speech turn with two speakers. Despite the creation of

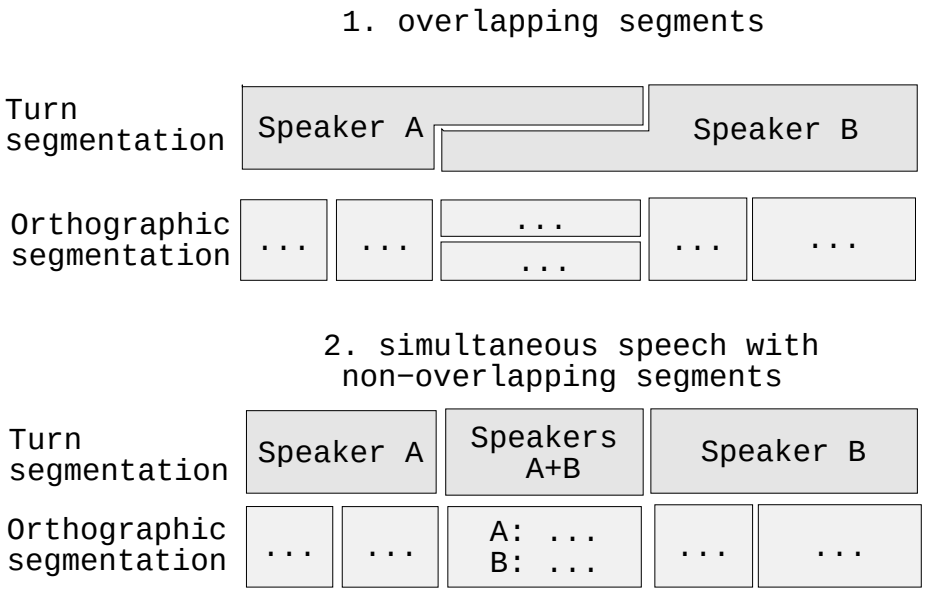

Figure 4: Two solutions tested for the representation of overlapping speech

this artificial speech turn, this led to a more acceptable solution in the interface. In the text editor, the parallelism between the two utterances has to be suggested as clearly as possible (cf. Figure 1).

In conversational speech, overlapping is often so common that this approach becomes problematic both for the transcriber and for the user. In the case of telephone speech recordings, two simultaneous speakers are often well enough separated on the separate channels for automated processing to go forward without special source-separation algorithms. In this case, it is much easier for the transcriber to segment and transcribe each channel as an independent stream, and the result is also more easily assimilated by training or testing programs as well as by human users. This approach to the transcription of heavily overlapped speech with a separate audio channel for each speaker (which is essentially the one that the LDC has been using) requires a different user interface as well as a different transcription specification. Providing such a solution in Transcriber

is one of our goals for the future. Meanwhile, we understand that one user has solved the problem temporarily by running two simultaneous invocations of Transcriber, one for each channel! The resulting files are then merged (or split) automatically later on. A better solution would integrate the parallel streams of transcription under simultaneous program control.

5 Practical use

Transcriber has been used for the DGA project on Broadcast News for over a year. It has also been used by the VECSYS French company for several months in the framework of the European Language Engineering project "OLIVE". In this section we describe the material which was transcribed, the working conditions and the productivity, the transcription guidelines which were provided to the transcribers, and report on the experience gained.

5.1 Material transcribed

The reference material for the DGA project was chosen from the national French program "France-Inter", and 20 hours taken from the morning news program (7h-9h) were recorded in December 1998 (10 weekdays from 2 consecutive weeks). This choice was motivated by the news-oriented but varied content, and by the fact that the distribution rights of this program could be obtained. The typical 2-hour program contains 3 news bulletins (for a total of about 50 minutes), specialized news (20 mn), various chronicles (10 mn), review of the French press and of the European press (15 mn), interviews and live questions from listeners (20 mn), and weather reports (5mn). The review of the European press is done by a non-native speaker, and contains of course a lot of foreign names and expressions.

The material transcribed by VECSYS were 15 hours of radio recordings from

French programs "France-Inter" and "France-Info", and 65 hours of television soundtracks from various channels in French and German (23 hours of "Arte" programs in French, 30 hours of "Arte" programs in German, and 12 hours of French channels "France 3", "France 2" or "TF1"). "Arte" programs consisted mostly in news bulletins and documentaries on social or political issues.

5.2 Working conditions

Two half-time working transcribers were hired for the DGA project. They were educated, native French speakers. Both were given a standard PC (Pentium Pro 200 MHz) under Linux with headphones and loud-speakers. Each one had to transcribe a set of 10 one-hour sound files copied to their hard disks. They worked in the same room and could share their experience. They had at their disposal dictionaries, newspapers from the period of the broadcast material to transcribe, and lists of journalist names. When they had completed a one-hour sound file, an additional verification was done in the presence of a researcher in order to discuss the specific problems which arose. Further normalizations were performed in the laboratory on the whole set of transcriptions.

Eight half-time working French or German native speakers produced the transcriptions for VECSYS. They started working within the company for about 15 days of training. Then they were provided with a PC running Linux with a modem and the sound files on a CD-ROM and worked at home. They were also given a list of known French journalists, and paper drafts which were available for some Arte programs; otherwise they relied on their own resources - some proved to do verifications using their internet access. The produced transcriptions were sent by e-mail. They were verified and corrected afterwards by a person specialized in this task and using all necessary dictionaries.

5.3 Productivity

A monitoring function was added to the tool in order to better analyze the production of transcriptions and estimate the amount of work needed for the transcription of one hour of material. This was also a user's request, since they were interested in monitoring their own daily progress. Time spent using the tool was measured and recorded, along with various measures on the transcription size (number of temporal breakpoints, of speech turns, of words...).

The total time needed for the production of one hour, including careful verification of the transcription, amounted to around 50 hours for both DGA transcribers. Of interest is that they did not follow the same strategy: the first one chose to segment and annotate the whole signal first, performing the orthographic transcription in a second pass; the second one did segmentation, annotation and transcription in parallel. The superiority of one strategy over the other one could not be demonstrated. However, it showed that much time was spent to get an accurate segmentation. This is an indication that a good automatic segmentation of the signal in short segments might speed up the overall transcription work.

Mean transcription time for the VECSYS experience amounted also to around 50 times real time, with a large disparity depending on the program. Radio news were easier, and television debate were much harder due to frequent overlapping speech and difficulty of speaker identification.

It was not possible to track the evolution of productivity of the transcribers, since the monitoring feature was not available at the beginning of the task and anyway the tool evolved much during the task. This would have to be studied using a stable version of the tool.

5.4 Transcription guidelines

Transcribers were provided a written document describing the transcription guidelines, i.e. explanations about what should be annotated and how to annotate it. Initial guidelines were written by LIMSI on the basis of previous ones that they had written for short utterances of spontaneous telephone speech and of their experience of broadcast news transcription. They were intentionally kept simple (and thus predictably incomplete) in their first version, and were augmented as necessary when specific questions arose.

The transcription guidelines covered the following topics:

- What should be annotated : orthographic transcription of the foreground; non-speech events and background noise conditions; speech turns with a precise identification of the speaker (name, gender, accent in the case of foreign speakers); topics. And what should not be annotated, such as transcription of commercials.
- How to add punctuation to increase readability without interfering with automatic processing.
- How to deal with numbers, spelled letters, unknown words, foreign words, etc.
- How to mark pronunciation errors, truncated words, overlapping speech.

Designing good guidelines proved to be far from being straightforward. They must meet several, sometime conflicting, requirements: they must ensure usability for several types of automatic processing, and take into account readability for the transcriber or another user who navigates through long transcriptions; they must help the transcriber in ambiguous situations and standardize the expected annotations, without bothering him with too many conventions which

might be difficult to remember or causing him to lose time on fine details; they must cover most cases without becoming inconsistent. To summarize, they have to keep a good balance between completeness and simplicity.

5.5 Experience gained

In practice, the initial transcription conventions have evolved along with the tool itself during the sessions to take into account the problems encountered and the transcribers' questions.

Capitalization

The principle was to transcribe mostly in lower case. Titles were also in lower case and put in quotes. Brand names, however, were capitalized. In practice, there were some confusions between titles and brand names, and both transcribers didn't always choose the same solution, e.g. for political party names. As a rule, following a standard orthography, e.g. as observed in the newspapers, should be the correct guess except when it makes automatic processing much harder. Designing specific entities - titles, brand names, etc. - should be left to another annotation mechanism independent of the orthography and could even be done automatically in some cases.

Acronyms

Transcribers were initially asked to prepend each acronym with a mark, depending upon it is read or spelled. This was thought to bring a potentially useful pronunciation information for a low additional cost. In practice, it brought errors in some non-ambiguous cases, and it was decided that a specific pronunciation mark was to be added only to acronyms for which both pronunciations are possible, and might even be dropped altogether.

Foreign words

In order to avoid processing utterances in foreign languages, these need to be marked. Foreign word or expressions in the transcription had to be marked with a language-switch tag, except for the most current foreign words which have become widely used in French. However, it was not always easy to decide if a word was current enough. Some foreign location names were rather frequent - e.g. "Wye Plantation" which is often mentioned because of the agreements signed there.

There were a lot of foreign expressions and names in the European press review which were difficult to transcribe. Transcribers were not requested to fully transcribe foreign expressions, but could do so if they wished.

Proper names

The transcribers at DGA always made impressive efforts to always find the correct writing of proper names, despite the fact that a specific marking was available for uncertain orthography. They were informed of the recording sessions that they will have to transcribe and decided to get newspapers from the corresponding days. This proved to be a valuable source for the orthography of the names. The name orthography of the journalists was available from the radio or in some specialized publications. Once again, the European press review proved to be a difficult challenge, since the foreign newspapers were not available to the transcribers, and they were often reluctant to mark a word as "orthography unknown".

Speaker identification

A specific situation was encountered by transcribers working on television soundtracks. Radio journalist are generally introduced, but this is not the case on

television. Speaker identification was much harder, and it was even difficult to gather the various occurrences of a same anonymous journalist across several programs.

Overlapping speech

Transcription of overlapping speech was of course a difficult point. Different situations were identified, each one with a different processing:

1. clear foreground speech with background speech - e.g. translation with the original foreign voice in background: in this case, only the foreground voice had to be transcribed with a noise marker indicating background speech.

2. limited interjections from other speakers (e.g. hum, yes...): they were indicated as instantaneous noises inside the main speaker transcription.

3. a dialog between two speakers with frequent overlapping at the boundaries: when feasible, it could be transcribed using the specific mechanism for simultaneous speech already described (cf. section 4.4).

4. more than two overlapping speakers: it was requested not to transcribe.

The 3rd situation occurred about 20 times per hour, with a mean extent of less than 3 seconds. The orthographic transcription of overlapping speech was difficult to produce; this is easily understandable since two sentences are intermixed, and the sound segment has to be listened to repeatedly. Locating the time boundaries was also difficult, since each speaker has its own starting and stopping times which generally occur inside a word from the other speaker. The user interface and data representation had to be modified several times before reaching an acceptable state. Even if it accounts for a few percent of the total

duration in our task, a correct management of overlapping speech should be carefully designed in any speech annotation tool.

As we indicated earlier, overlapping in conversational speech, especially where each speaker is recorded on a separate audio channel, typically motivates a different approach in which each speaker's utterances and pauses are treated as a separate partition of the time stream, with the relationship between speakers' productions determined by the relationship of these logically independent time markings.

6 Future directions

6.1 New needs

The need for several new features was identified during our experiments:

- More tools are clearly needed for ensuring the consistency of proper names throughout the various transcriptions. A user-defined glossary and editable shortcuts have been introduced in the tool at user request; however, this is not yet completely satisfactory. Online dictionaries, encyclopedias, or even maps for place names, should be made easily available to the transcriber. A mechanism of automatic completion using previously written names in all existing transcriptions (compiled by hand or even automatically) seems to be an interesting solution and remains to be implemented.

- Speaker identification on television soundtracks was very difficult. The best solution for this problem would be to provide the complete video recording, not only the audio track. This would also ease the whole transcription process in the case of background noise. With the current development of video capabilities on standard computers, it can be hoped that easy technical solutions for interfacing the tool with a video player will be

available in the near future. Such an interface will also be useful for other applications, such as the study of gesture in communicative interaction. In the short term, watching the video during the verification phase is an alternative (as has been the practice at the LDC).

- An automatic segmentation could be given to the transcriber as a starting point in order to speed-up the segmentation phase.
- Multiple sound files could be conveniently be managed in a single transcription file. Also, a specific management should be available for multi-channel sound files (e.g. telephonic conversations as in the Switchboard task).

Some limitations remains. Management of transcription files over 1 hour becomes slow in our configurations, mostly because of the numerous embedded buttons inside the text editor which reaches the limits of the Tk library; at the same time, signal display remains perfectly smooth. Also, the "undo" function is too limited.

6.2 Format evolution

Concerning the data format, XML is a convenient way to structure a complex document, but the design of our DTD was in much ways arbitrary. We kept from previous LDC formats and from UTF a transcription structured in a single tree, which brought serious limitations to further extensions.

Future developments will without any doubt take into account S. Bird and M. Liberman reflections about annotation graphs [2][3]. They show that virtually any existing annotation can be viewed as a labelled acyclic graph, some nodes of which bear ordered time values, and they develop a complete formalism for annotation graphs. Within this framework, all segments of the transcriptions

would be stored as an unordered set of typed arcs between identified nodes.

Switching to this framework for internal data management and for the reference transcription file format would lead to a much more generic tool, and conversion to other formats would become easier. This does not prevent us from providing alternative formats, with time-ordered segments or in a human-readable format. The format would no longer constrain speech turns to be contiguous, or new sections to impose a new turn, though such constraints could remain in the interface of the tool itself if wanted.

7 Conclusions

Interface prototyping in a scripting language proved to be an efficient development way, provided robust libraries are available. Being distributed as free software, our project has been followed by numerous speech scientists and engineers who gave precious hints for further developments and made the tool much more portable and usable. A web site has been designed for the distribution of the tool, and an announcement and a developer mailing list are already in function [16]. Transcriber is now used by several research or development teams in various countries. Most effort was devoted to development, and external contributions were not organized despite of spontaneous proposals; but future versions should be developed in a modular fashion with an interactive dialog with the potential co-developers. Future developments will be based on recent reflections about annotation graphs for a more general approach.

After more than one year of experience, we feel that Transcriber is suitable for large-scale production of speech resources. It is not universal, and other tools exist [4]; among them the segmentation tool developed at ISIP shares several conception choices with our tool [8]. But we learned much from its development, and we hope it can benefit to the whole community.

Acknowledgements

The initial directions of the tool and the most promising ideas for its evolution arose from discussions with LDC members, especially Mark Liberman. Initial transcription conventions for French were designed by Martine Adda from LIMSI. Martine Garnier-Rizet coordinated the use of Transcriber within VECSYS and gave us a valuable feedback on this. Snack's developer Kåre Sjölander was very helpful in always taking into account the changes which were needed for Transcriber. We are also glad to thank here all users and testers who gave us report of their experience and their problems, since this helped us very much improving our tool. And finally many thanks to the transcribers for their patience using a program under development!

References

[1] C. Barras, E. Geoffrois, Z. Wu, M. Liberman, "Transcriber: a Free Tool for Segmenting, Labeling and Transcribing Speech", *Proc. 1st Int. Conf. on Language Resources and Evaluation*, pp. 1373-1376, Granada, May 1998.

[2] S. Bird, M. Liberman, "A Formal Framework for Linguistic Annotation", *Technical Report MS-CIS-99-01*, Department of Computer and Information Science, University of Pennsylvania, 1999.

[3] S. Bird, M. Liberman, "Annotation graphs as a framework for multidimensional linguistic data analysis", *Proceedings of ACL Workshop: Towards Standards and Tools for Discourse Tagging*, pp. 1-10, 1999.

[4] S. Bird, M. Liberman, "Linguistic Annotation", `http://www.ldc.upenn.edu/annotation/`.

[5] F. Bonnet, "tcLex: a lexical analyzer generator for Tcl", http://www.multimania.com/fbonnet/Tcl/tcLex/index.en.htm .

[6] *Document Object Model (DOM)*, http://www.w3.org/DOM/.

[7] *Extensible Markup Language (XML)*, http://www.w3.org/XML/.

[8] *ISIP software*, http://WWW.ISIP.MsState.Edu/projects/speech/software/.

[9] M. K. McCandless, *A Model for Interactive Computation: Applications to Speech Research*, Ph.D. Thesis, Massachusetts Institute of Technology, 1998.

[10] J. K. Ousterhout, *Tcl and the Tk Toolkit*. Addison Wesley, ISBN: 3-89319-793-1, 1994.

[11] J. K. Ousterhout, "Scripting: Higher-Level Programming for the 21st Century", *IEEE Computer*, March 1998.

[12] K. Sjölander, "The Snack Sound Visualization Module", http://www.speech.kth.se/snack/.

[13] K. Sjölander, J. Beskow, J. Gustafson, E. Lewin, R. Carlson, B. Granström, "Web-based Educational Tools for Speech Technology", *Proc. 5th Int. Conf. on Spoken Language Processing*, pp. 3217-3220, Sydney, November 1998.

[14] *The Tcl/Tk scripting language*, http://www.scriptics.com/.

[15] *TEI Guidelines for Electronic Text Encoding and Interchange (P3)*, http://www.uic.edu/orgs/tei/.

[16] *Transcriber*, http://www.etca.fr/CTA/gip/Projets/Transcriber/.

Contents