Semi-Automatic Transcription of Interviews

Thomas Lüdi

Semester Thesis
May 2014

Superviser
Dr. Beat Pfister

Adviser
Prof. Dr. Lothar Thiele

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

TIK
Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Abstract

Description of a system that automatically generates EXMARaLDA-Partitions for selected audio files. It uses MacSpeech Scribe to transcribe audio files and a compiled Matlab function to calculate timing information for the transcribed text.

Acknowledgement

I would like to thank my superviser Dr. Beat Pfister for his support and suggestions during this project. Especially his insight on how a user might use the application was a great help.

Contents

1

Introduction

To analyze differences in the learning progress between german and french speaking children, interviews displaying their proficiency in the french language are conducted. These interviews can take up to an hour and the transcription into a format that can be analyzed is very work intensive. An automatic transcription, even if only the questions of the interviewer can be transcribed, would therefore be very helpful, as it helps navigating the file to find the answers of the children.

A helpful tool for the analyzation of the interviews is EXMARaLDA, the EXtensible MARkup Language for Discourse Annotation. This tool needs the start and end time of transcribed phrases. Unfortunately, most transcription software only outputs the transcribed text without the timing information. The system described in this work will automatically transcribe a selection of interviews from mp3-files, calculate the necessary timing information and generate the EXMARaLDA-Partitions

2

Components

2.1. Applescript

AppleScript is a scripting language that provides direct control of scriptable applications and scriptable parts of the Mac OS. A scriptable application is one that can respond to a variety of Apple events by performing operations or supplying data. An Apple event is a type of interprocess message that can encapsulate commands and data of arbitrary complexity. By providing an API that supports these mechanisms, the âĂIJOpen Scripting ArchitectureâĂİ makes possible one of the most powerful features in OS XâĂŤthe ability to write scripts that automate operations with multiple applications.

You can use AppleScript scripts to perform repetitive tasks, automate complex workflows, control applications on local or remote computers, and access web services. Because script writers (or scripters) can access features in any scriptable application, they can combine features from many applications. For example, a script might make remote procedure calls to a web service to get stock quotes, add the current stock prices to a database, then graph information from the database in a spreadsheet application. From controlling an image-processing workflow to performing quality assurance testing for a suite of applications, AppleScript makes automation possible.

While the AppleScript scripting language (described in AppleScript Language Guide, and in a number of detailed third-party books) uses an English-like terminology which may appear simple, it is a rich, object-oriented language, capable of performing complicated programming tasks. However, its real strength comes from providing access to the features available in scriptable applications. If you make your application scriptable, it will help scripters get their work done, and quite likely become indispensable to their work process.

The âĂIJAutomatorâĂİ application, available starting in OS X version 10.4, lets users work in a graphical interface to put together complex, automated workflows. Workflows consist of one or more actions, which are provided by Apple, by developers, and by scripters, and can be written in AppleScript and in other languages, including Objective-C. Starting in OS X v10.5, developers can incorporate workflows directly in their applications, providing another mechanism for accessing features of other applications and the Mac OS. âĂIJScripting Bridge,âĂİ available starting in OS X version 10.5, provides an automated process for creating an Objective-C interface to scriptable applications. This allows Cocoa applications and other Objective-C code to efficiently access features of scriptable applications, using native Objective-C syntax. Some other scripting languages, such as Ruby and Python, can use Scripting Bridge, but also have their own software bridges to access features of scriptable applicationsâĂŤfor more information, see Getting Started With Scripting and Automation.

AppleScript has several other new or improved features in OS X v10.5, including full support for Unicode text, additional support for identifying and working with application objects in scripts, 64-bit support, more accurate and useful error messages, and additional scriptability in Apple technologies such as iChat and the Dock. For more information, see AppleScript Features.[1]

2.2. EXMARaLDA

EXMARaLDA is an acronym of "Extensible Markup Language for Discourse Annotation". It is a system of concepts, data formats and tools for the computer assisted transcription and annotation of spoken language, and for the construction and analysis of spoken language corpora. EXMARaLDA was originally developed in the project "Computer assisted methods for the creation and analysis of multilingual data" at the Collaborative Research Center "Multilingualism" (Sonderforschungsbereich "Mehrsprachigkeit" - SFB 538) at the University of Hamburg. Since July 2011, the development of EXMARaLDA is continued at the Hamburg Centre for Language Corpora, since November 2011 in cooperation with the Archive for Spoken German at the Institute for the German Language in Mannheim. All components of the EXMARaLDA system are freely available. The main features of EXMARaLDA are:

- XML based data formats All EXMARaLDA data are stored in XML files. The use of this W3C standard ensures flexible usability and long-term archivability of the data.
- Java based tools All software tools for creating and working with EXMARaLDA data (Partitur Editor, Corpus Manager and Query tool EXAKT) are JAVA applications. This makes them suitable for all currently used operating systems (Windows, Macintosh, Linux, Unix).
- Interoperability The EXMARaLDA concept is based on the annotation graph framework (Bird/Liberman 2001) and thus aims at a maximal exchangeability and reusability of transcription data. Hence, it is possible to create and edit EXMARaLDA data not only with the system's own tools, but also with other popular software (like Praat, ELAN,

[1] https://developer.apple.com/library/mac/documentation/applescript/Conceptual/AppleScriptX/Concepts/ScriptingOnOSX.html#//apple_ref/doc/uid/20000032-BABEBGCF May 2014

Transcriber or FOLKER).

Furthermore, EXMARaLDA data can be transformed into a number of widely used presentation formats (RTF, HTML, PDF) for web-based or printed publication. Last but not least, EXMARaLDA supports several important transcription systems (HIAT, DIDA, GAT, CHAT) through a number of parameterised functions.[2]

2.3. MacSpeech Scribe

MacSpeech Scribe is a speech recognition software built for the Mac OS X. It transcribes recorded voice files of type .aif, .aiff, .wav, .mp4, .m4a and .m4v. It is trained to a single voice and the used vocabulary can be adjusted for the expected content. It is owned by Nuance Communicatinos. Nuance also developed other speech recognition sofware like Dragon NaturallySpeaking for Windows and Dragon Dictate for Mac.[3]

2.4. Audio Hijack Pro

Audio Hijack Pro is an audio capturing software that records audio directly from an application or from hardware like microphones or speakers. It can record to a number of different formats like mp3, aac, aiff and wav. It can add tags and effects to the recorded audio and it can be scheduled to record at a given time. It is developed and distributed by rogue amoeba.[4]

2.5. MATLAB

MATLAB is a fourth-generation programming language developed by MathWorks. It incorporates a high-level technical computing lanugage and interactive environment for algorithm development, data visualization, data analysis and numerical modeling. With the programming language, the tools and the integrated mathematical functions, different approaches can be tested to get a solution faster than with conventional programming languages like C/C++ or Java.[5]

In this project the function mp3read[6] developed by Dan Ellis was used to read mp3 audio files. It includes binaries from the LAME Project.[7]

[2] http://www.exmaralda.org May 2014

[3] http://www.nuance.com/for-individuals/by-product/dragon-for-mac/index.htm May 2014

[4] https://www.rogueamoeba.com/audiohijackpro/ May 2014

[5] http://www.mathworks.ch/products/matlab/ May 2014

[6] http://www.mathworks.com/matlabcentral/fileexchange/13852-mp3read-and-mp3write May 2014

[7] http://lame.sourceforge.net/ May 2014

3

EXMARaLDA Transcriber

3.1. Transcribing Files

The main part of the EXMARaLDA Transcriber is the transcription of interviews and following generation of the EXMARaLDA-Partitions.

3.1.1. Scripting MacSpeech Scribe

Since MacSpeech Scribe does not support scripts, alternate approaches are necessary to automate it. The first problem that appears is a version warning, that appears if MacSpeech Scribe is run on Mac OS X 10.8. Applescript can not check whether this window has appeard and must therefore wait the worst case time until it can perform any further action. Despite not having direct access to the buttons, using Direct Keyboard Access we can switch between them and press one using simulated keystrokes. The script then checks the name of MacSpeech Scribe's front window every second until the profiles window appears.

Since the voice profiles can only be selected by mouse and there is no other possibility to load load a profile, the user is prompted to select a it himself. The user is then asked to select the audio files. The script allows multiple selections of files of the type .mp3 and .wav. Lastly the user has to select an output folder, where the EXMARaLDA-Partitions will be generated. The 'Make Active' button of the profiles window can not be accessed by the script directly, nor by the Full Keyboard Access feature of the Mac Os. We use a simulated mouse click to continue. Hard coded coordinates for the mouse click would not work in different screen resolutions, or if the window has been moved, so we calculate the buttons position relative to the window's position and size.

Before an audio file can be loaded by MacSpeech Scribe when the next window appears, a dummy operation in the form of a button press must be performed, or MacSpeech Scribe will crash. Luckyly this window is accessible by Applescript and the 'Transcribe' button can be activated directly by the script. The window to select a file can also be canceled by the script directly and MacSpeech Scribe returns to the previous window. At this point, MacSpeech scribe is fully started up and ready to accept audio files. MacSpeech Scribe does not support mp3-files so if the selected audio file is in the mp3 format a compiled Matlab function is called to create a wav-file from it.

This wav-file is then loaded into MacSpeech Scribe with a shell script. In the appearing window the 'Transcribe' button again needs to be clicked with a simulated mouse click and again we use the window's position and size to get the coordinates. The next window opens and the transcription starts. Since the script can not check directly whether MacSpeech Scribe has finished transcribing, it tries to select the text, which can only be selected when the transcription has finished, with simulated keystrokes.

3.1.2. Acquiring Audio Using Audio Hijack Pro

MacSpeech Scribe does not output any timing information about the transcribed text, howewer it can play the audio belonging to a selection of words that it beliefs belong to the same phrase. These phrases can be selected and the corresponding audio played by hitting the keyboard shortcuts with simulated keystrokes. The audio is then recorded by Audio Hijack Pro. To check whether MacSpeech Scribe is still playing sound, the script checks if the recording time is still going up, since it stops if there is no audio to record. The recording is done in the wav-format with a sample rate of 22050 samples per second. The resulting files are stored in the temp folder of the EXMARaLDA Transcriber directory. They are named the same as the source file with the number of the phrase attached.

At the same time, the selected phrases are copied by simulating the keystrokes ctrl+c and stored in an Applescript variable. Since the script intern function to copy the clipboard caused crashes sometimes, the shell script function 'pbpaste' was used instead.

When the last phrase is selected, trying to select the next phrase will stay on the current one. When the same phrase is copied three times in a row, the script assumes it has reached the end of the file and starts the Matlab function to compute the timing information. Because Matlab input strings can not contain a space when called with a shell script, all spaces in the path to the source audio file are replaced by a special character not appearing in the string when passed as an input. This character is then switched back to a space in the Matlab function.

3.1.3. Getting Timing Information using Matlab

To get the start and end time of every phrase, the audio of the phrases is cross correlated with the source audio file. To speed up this process, only a small part of the source audio file is used for correlation. The part that gets selected starts from half a second before the end of the previous phrase and is three times the length of the current phrase plus one second long. From the index of the highest correlation value, the start and end position of the phrase is calculated. If the correlation value scaled to the length of the current phrase is too low, the result is considered invalid and the next part of the source file is selected for correlation. If after twenty parts, no valid result was found, the part that got the highest correlation value is considered the correct one, howewer the next phrase will use the end time of the phrase before the last one to calculate the current part of the source audio file used for correlation.

This process is done with 50 phrases at a time, so some progress feedback can be given to the user. When the timing information for the phrases has been calculated, the start and end times in milliseconds and the phrase number are written to a file. If the first phrase of the transcription is in the current 50 phrases, the total length of the source audio file is also written to the start of the file. With the timing information in this file and the phrases stored in the Applescript variable the EXMARaLDA-Partition can be generated.

3.1.4. Generating the EXMARaLDA-Partition

The EXMARaLDA-Partition uses the xml-format. It contains a constant Framework, the relative path to the source wav-file, the phrases and their start and end times. To account for faulty recordings, the phrase numbers of the text and timing information are compared. If timing or text information for a phrase is missing, it will not be written to the EXMARaLDA-Partition. When the file is written and stored in the output folder, the phrase audio and the timing information file are deleted, the MacSpeech Scribe transcription window is closed and the next audio source file is loaded for transcription.

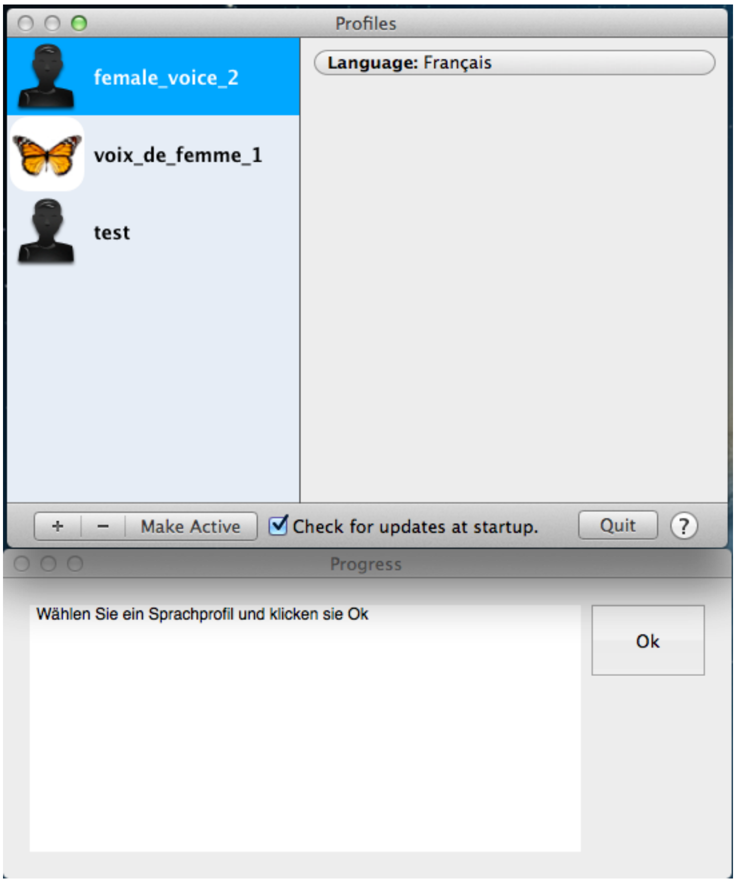
3.1.5. Feedback to the User

To give feedback to the user about the progress of the transcription, the EXMARaLDA Transcriber displays a small window that shows which file is currently being handled and which part of the process is running at the moment. This information can be displayed in english or german, which can be selected during installation.

3.2. Additional Functions

3.2.1. Free Disk Space

If a source file used for a transcription is in the mp3 format, a wav-file has to be generated for EXMARaLDA to work properly. Unfortunately a wav-file uses much more disk space than a mp3-file. When starting up EXMARaLDA Transcriber a function to reduce disk space usage can be called. It deletes the wav-file associated with the EXMARaLDA-Partition and changes the EXMARaLDA extention .exb to .exm so it is not accidentially called when the source file is missing.

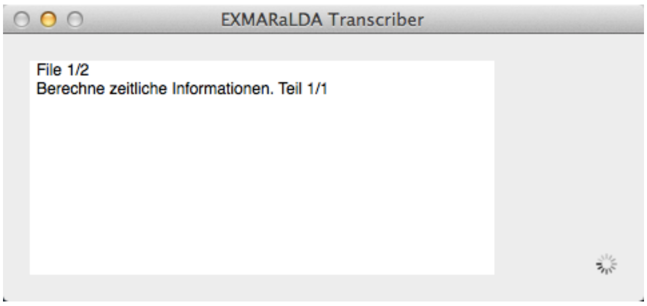


Language: Français

☑ Check for updates at startup.

Wählen Sie ein Sprachprofil und klicken sie Ok

Figure 3.1.: Prompting the user to select a voice profile



File 1/2
Berechne zeitliche Informationen. Teil 1/1

Figure 3.2.: Giving progress feedback

3.2.2. Reconstruct Wav-Files

When starting up EXMARaLDA Transcriber a function can be called to restore an EXMARaLDA-Partition that had its source wav-file deleted to working order. It generates the wav-file from the mp3-file with the same name as the source wav-file indicated in the EXMARaLDA file. Afterwards the extention is reverted to .exb so the file can be opened by EXMARaLDA again.

4

Conclusion and Outlook

EXMARaLDA Transcriber is an application that takes a voice profile, audio files of the type mp3 or wav, and an output folder as input and outputs the EXMARaLDA-Partitions for the input files in the output folder without requiring further user input.

4.1. Further Work

Further additions to EXMARaLDA Transcriber could include support for more input file types like mp4 or aif. A useful addition would also be a pause and/or quit button for better controllability by the user.

A

Appendix

A.1. Handbook

A.1.1. Introduction

EXMARaLDA Transcriber is a tool to automatically transcribe audio files and create EXMARaLDA-Partitions. It uses MacSpeech Scribe to transcribe audio files and calculates timing information for EXMARaLDA using phrases recorded with Audio Hijack Pro and cross correlation functions from MATLAB.

A.1.2. Installation

- Start setup.app on the install-cd.
- Choose the operating language of EXMARaLDA Transcriber.
- Choose the install folder.

Necessary System Settings

Enable Full Keyboard Access: System Preferences > Keyboard > Full Keyboard Access: All controls

A.1.3. General Operation

- Start the application EXMARaLDA Transcriber.
- Choose the task you wish to perform
  1. Transcribe File: Generates EXMARaLDA-Partitions for selected audio files.
  2. Free Disk Space: Frees up disk space by deleting wav-files generated from mp3-files. The corresponding EXMARaLDA-Partition can not be used without the wav-file.
  3. Restore Files: Restores wav-files for EXMARaLDA-Partitions so they can be worked on again.

Transcribe File

- If MacSpeech Scribe is running already, you will be asked to quit it. MacSpeech Scribe must be started internally for EXMARaLDA Transcriber to work properly.
- Wait until the first startup phase has finished. You will be prompted to select a voice profile.
- Select a voice profile and click 'Ok' in the window 'Progress'.
- Choose audio files to be transcribed. You can choose multiple files with the command-key or the shift-key.
- Choose the output folder. See recommended folder structure
- The transcription will start now. Please do not use the keyboard or the mouse during this phase.
- If an input file is of the type mp3, a wav-file will be generated.
- The audio files are transcribed and separated into phrases. These phrases are recorded and the start and end times are calculated. During this time, the system volume will be muted. At the end the EXMARaLDA-Partition is generated and stored in the output folder.
- When all the transcriptions are finished, click 'Quit' to close EXMARaLDA Transcriber.

Free Disk Space

Since the audio source for EXMARaLDA needs to be a wav-file to get all the functionalities, a wav-file is generated if the original audio file was an mp3-file. Wav-files require much more disk space than mp3-files. To free up this space, EXMARaLDA Transcriber deletes the wav-file and changes the extention on the EXMARaLDA file to .exm so it is not accidentally opend.

Warning: Do not delete the mp3-file or the EXMARaLDA file can not be recovered

To free up disk space, select the .exb files that are not currently used when you are asked to.

Restore Files

Choose this function to recover files whose wav-files have been deleted. The wav-file will be regenerated from the mp3-file and the exm-files will be restored to EXMARaLDA files.

To recover files, select the .exm files that you wish to recover when you are asked to.

A.1.4. Create or Improve Voice Profile

For more information please refer to the manual of MacSpeech Scribe or click the ? button in the profiles window of MacSpeech Scribe.

Create Voice Profile

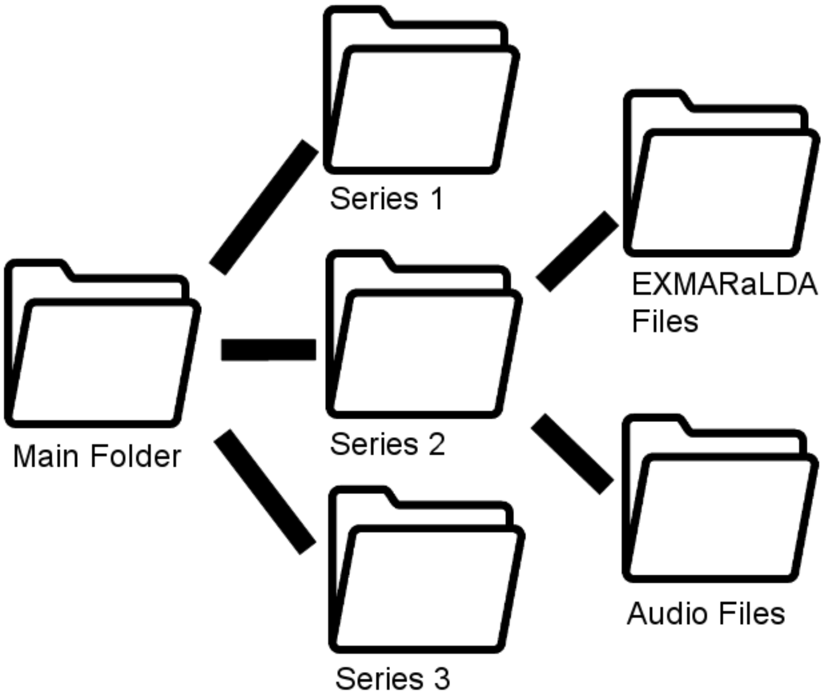
- Start MacSpeech Scribe.
- If an operating system warning appears, click 'Continue'.
- Click the + button in the bottom left corner of the profiles window.
- Follow the instructions of MacSpeech Scribe.

Improve Voice Profile

- Start MacSpeech Scribe.
- If an operating system warning appears, click 'Continue'.
- Choose the voice profile you wish to improve.
- Click 'Make Active'.
- Click 'Transcription Tranining'.
- Follow the instructions of MacSpeech Scribe.

A.1.5. Recommended Folder Structure

To relocate EXMARaLDA files, the relative path to the audio source needs to be preserved. The folder structure shown below allows relocating the series folders as a whole without compromising the functionality.


Series 1
Main Folder
Series 2
Series 3
EXMARaLDA Files
Audio Files