Automatic Annotation of Media Field Recordings

Peter Wittenburg, Eric Auer, Han Sloetjes,[1] Oliver Schreer, Stefano Masneri,[2]
Daniel Schneider, Sebastian Tschöpel[3]

Abstract. In the paper we describe a new attempt to come to automatic detectors processing real scene audio-video streams that can be used by researchers world-wide to speed up their annotation and analysis work. Typically these recordings are taken in field and experimental situations mostly with bad quality and only little corpora preventing to use standard stochastic pattern recognition techniques. Audio/video processing components are taken out of the expert lab and are integrated in easy-to-use interactive frameworks so that the researcher can easily start them with modified parameters and can check the usefulness of the created annotations. Finally a variety of detectors may have been used yielding a lattice of annotations. A flexible search engine allows finding combinations of patterns opening completely new analysis and theorization possibilities for the researchers who until were required to do all annotations manually and who did not have any help in pre-segmenting lengthy media recordings.

1 BACKGROUND

Many researchers in linguistics such as field workers and child language researchers have to work with real scenario sound and video material. Field recordings are often more challenging to process than lab recordings, for example for pattern recognition tasks. The reasons for this are manifold such as inadequate and varying position of the sensor devices (microphone, camera), various types of background noise, the need to use consumer grade devices etc. Standard speech and image recognition techniques only deliver very poor results for such recordings. Of course there are also many resources with better recording quality, but they often involve non-standard languages, long stretches of silence or regular patterns resulting from experimental settings etc. Yet, annotators would like to use any help they can get to make their work more efficient, because manual annotation is so time consuming.

There is often little knowledge about the analyzed languages, so we miss formal descriptions such as proper language models. The consequence is that researchers who want to analyze this sort of material need to first carry out manual annotations based on time consuming listening and watching. In 2008, we made statistics amongst 18 teams documenting endangered languages within the DoBeS[4] program to find out how much time is required for the most essential workflow steps. According to these statistics creating a transcription costs 35 times real-time (i.e. a transcription of an one-hour video requires at least 35 hours), a translation into a major language 25 times real-time and for any special linguistic encoding such as morphosyntactic glossing or gesture annotation the costs in general are much higher than 100 times real time.

Because annotating is so time-consuming, an increasing number of recordings in the archives of the Max Planck Institutes are not annotated and even not touched any more, i.e. valuable material cannot be included in analysis of the linguistic system, theoretical considerations and cultural and cognitive studies. Advanced annotation and analysis tools such as ELAN[5] and ANNEX[6] can facilitate the difficult work and can speed up the process slightly although no quantitative factors can be given. Yet these tools do not operate at the content level of the media streams.

2 DESIGN CONSIDERATIONS

Motivated by this unsatisfying development some brainstorming between researchers and technologists of two Max Planck Institutes on the one side and sound and image processing specialists from two Fraunhofer Institutes was initiated to discuss ways out leading to a three year innovation project funded by Max Planck Gesellschaft and Fraunhofer Gesellschaft. Actually an old idea spelled out in the Hearsay II system [1] was brought into consideration again. In Hearsay II more or less complex independent knowledge components were operating on the speech signals each of them writing their findings on a blackboard. Other knowledge components were added that analyzed the blackboard findings to finally create an automatic transcription of what was said. Such a knowledge based architecture has the potential of being used to let the user interact with the low level audio and video analysis components, which was one of the major requirements of the researchers at the Max Planck Institutes participating in this innovation project.

In AVATecH[7], detector components analyze audio or video input streams and generate annotations or intermediate results. Detectors can use the output of other detectors as input, in addition to the audio and video source files.

After having analyzed a preliminary evaluation corpus with a variety of recordings provided by the Max Planck Institutes, we found that the characteristics of the data are indeed challenging for acoustic analysis. 55 scenes from about 30 files include wind noise and similar, about 10 have reverb, about 15 considerable background noise (engines, people, etc.) and 5 contain humming sounds. About 20 scenes seem to be not useful for any type of audio analysis. The speech quality itself is also varying from "indistinguishable talking"

[1] Max Planck Institute for Psycholinguistics, Nijmegen, the Netherlands, email:peter.wittenburg@mpi.nl

[2] Fraunhofer Heinrich Hertz Institute, Berlin, Germany

[3] Fraunhofer Institut für intelligente Analyse- und Informationssysteme IAIS, Sankt Augustin, Germany

[4] Dokumentation Bedrohter Sprachen, www.mpi.nl/dobes/

[5] Eudico Linguistic Annotator, www.lat-mpi.eu/tools/

[6] A web-based annotation viewer, www.lat-mpi.eu/tools/

[7] Advancing Video Audio Technology in Humanities, www.mpi.nl/research/research-projects/language-archiving-technology/avatech/avatech-project/

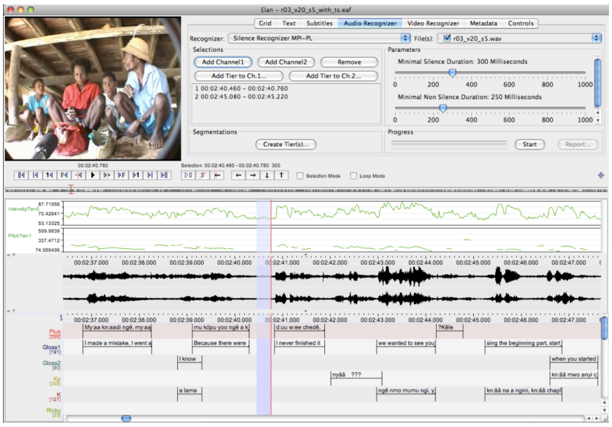
to intelligible speech. The results of acoustic segmentation, speech detection, speaker clustering and gender detection with standard algorithms optimized for broadcast data were rather disappointing as was expected. Due to the variety of languages, classic mono-lingual speech recognition could not be applied.

The initial corpus analysis resulted in a number of conclusions:

- return to the blackboard type of scenario where "detectors of various sorts" will create annotations on a new specific tier
- start experimenting with so-called low hanging fruits, i.e. simple detectors that can be integrated quickly based on existing algorithms
- have smart search and filtering methods to allow researchers to easily browse through (complex) annotation lattices
- allow the researcher to interact with the annotations and easily modify parameters controlling the functioning of the detectors so that manual tuning can be used instead of using a "one size fits all" stochastic method
- rely on existing technologies where possible with respect to the annotation and search framework and the pattern detectors

3 ANNOTATION AND SEARCH FRAMEWORK

ELAN is currently one of the most widely used media annotation tools in various linguistic sub-disciplines and beyond. It allows researchers to hook up an arbitrary number of annotation tiers referencing custom vocabularies to multiple media streams that share the same timeline. The fact that annotations cannot only be attached to a time segment but also to annotations on other tiers provides support for the creation of complex annotation structures, such as hierarchical annotations trees.



Recognizer: Silence Recognizer MPI-PL

File(s): ☑ r03_v20_s5.wav

1 00:02:40.460 - 00:02:40.760
2 00:02:45.080 - 00:02:45.220

Minimal Silence Duration: 300 Milliseconds

Minimal Non Silence Duration: 250 Milliseconds

Selection: 00:02:40.460 - 00:02:40.760 300

Figure 1. Use of a silence detector in ELAN 3.6: Detector parameters can be adjusted on the right side. A video viewer is on the left side. Results will be added as a new tier to the annotations and waveforms at the bottom.

In contrast to comparable tools [2] such as Frameline 47, ANVIL, EXMARaLDA or Advene[8], ELAN's advantages include an open-source core, unlike the commercial Frameline 47 or closed-source ANVIL. This is ideal for extending the tool with detection algorithms. Also, ELAN already supports numerous import and export formats (in contrast to EXMARaLDA or Advene) relevant for linguistic research such as PRAAT, Chat, Shoebox or Transcriber data. Like most of the tools mentioned, ELAN is platform independent: It is available for Mac OS, Windows and Linux.

The underlying EAF (ELAN Annotation Format) schema emerged from the early discussions about models such as Annotation Graph [3] and it is flexible enough to cater for a large number of tiers with variable vocabularies being created by a number of (small) detectors. The screenshot in figure 1 depicts a typical ELAN window layout. ELAN has many functions including the possibility to start the well-known PRAAT[9] speech analysis software for a specific, detailed acoustic analysis.

ELAN is accompanied by TROVA[10], a flexible search engine that allows users to search for complex annotation patterns within annotation tiers, across several annotation tiers, over time and/or annotation sequences. Each pattern can be specified as a regular expression offering a large degree of flexibility. TROVA operates not only on the visualized resource, but can be used to operate on a whole selection of resources resulting from metadata searches or composed by the user. Using indexes created at resource upload or integration time, TROVA can operate very fast on large amounts of data. While the user reads the first results, TROVA continues to search further matches in the background when searching in a large corpus.

The current tools are an excellent starting point for improvements in the direction of adding new semi-automatic annotation and extended search functionality. Also, users are already familiar with the user interfaces, making it easy for them to adopt new functionality.

4 FIRST INTEGRATION EXAMPLE

The first recognition component that was integrated as a test case offers simple detection of pauses (silences) in sound recordings – in fact a well-studied detection problem, the potential errors of which are known. The user graphically configures the essential parameters and receives a graphical indication of the usefulness of the choices immediately after execution. This feature of ELAN is already applied by a variety of users and it speeds up their work considerably. Some of the scenarios are:

- In experiment result analysis, users want to quickly index or remove periods of silence in order to reduce the length of the sound wave to be analysed to a minimum.
- Field linguists want to use the "annotation step through" function of ELAN to quickly navigate from one sequence of speech to the next, thus carrying out a first very rough selection of the material.
- Gesture researchers can now more easily create statistics that interrelate the timing of gesture and speech segments.

It is not solely the complexity of the detection function that counts; in this particular low-hanging fruit example it is the packaging into a tool such as ELAN and the convenient graphical interaction that are attractive to researchers. The typical errors produced by such detectors are in general not dramatic, since the researchers likely use the detected segments either just for quick inspection or as a base segmentation that might be manually corrected and extended. A complete API for plug-ins or components being executed on remote servers has been worked out and has been verified. This API is documented in a manual which is available online [4].

[8] www.frameline.tv/ – www.anvil-software.de/ – www.exmaralda.org/ – http://liris.cnrs.fr/advene/index.html/

[9] www.fon.hum.uva.nl/praat/

[10] Annex / Trova: www.lat-mpi.eu/tools/annex/ or select "annotation content search" in the IMDI Browser http://corpus1.mpi.nl/

5 LOW HANGING FRUIT DETECTORS

Currently a number of such low-hanging fruit detectors have been studied on test corpora and are being integrated into the ELAN framework. For audio signals we are working on robust audio segmentation, speech detection, speaker clustering and pitch contour detection. For video, we are working on the integration of shot and sub-shot boundary detection, motion recognition (camera and scene motion), face detection and tracking of body parts. We also investigate possibilities for gesture recognition.

5.1 Segmentation

Noise-robust segmentation of the audio stream into homogeneous segments inserts boundaries e.g. between speakers or at other significant acoustic changes. The algorithm will be capable of providing fine-grained segmentation of speaker utterances [5]. The user can control the granularity of segmentation by tuning a corresponding feedback parameter.

5.2 Speech detection

This detector finds audio segments which contain human speech, in a language-independent way. Naturally, weak audio quality is a drawback for the detection quality. Furthermore the various research recordings are very heterogeneous. Thus, we enable the user to manually annotate a small amount (less than five minutes) of non-speech segments in order to adapt the model to the given data which leads to a more robust detection.

5.3 Speaker clustering

A language-independent intra-document speaker clustering algorithm labels identical speakers within a single document with the same ID (see [6], [7] and [8]). The results can be used for removing the interviewer in a recording, or for extracting material from specific speakers from a recorded discussion. For optimization of the detection performance we use manual user input, e.g., the number of speakers or speaker audio samples.

5.4 Vowel and pitch contour detection

The pitch contour detector can allow researchers to graphically specify typical pitch contours and search for similar patterns. We already implemented a detector which tags vowel segments in audio recordings and annotates the corresponding time-spans with pitch and intensity properties such as for example minimum, maximum, initial or final f0 frequency, or volume. The detector invokes PRAAT to calculate f0 and volume curves of the input over time. Those are then used to find characteristic segments and annotate them.

5.5 Shot and sub-shot boundary detection

The shot and sub-shot boundary detector (see [9], [10] and [11]) identifies scene changes as well as considerable changes in the video scene. Since different shots refer to different camera operations, all the subsequent detectors work on a shot basis. Each detected scene as well as scene changes are marked by a still frame, in order to represent all of the content in the video and allow the user to browse through it without actually watching the video. The detector processes about 80 frames every second on a single core 3.6 GHz Pentium IV, i.e. an hour of video is processed in less than 20 minutes. An example of the results from this detector is shown in figure 2.

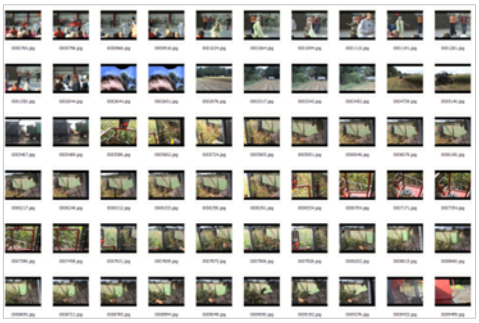


Figure 2. This figure shows the results of a shot and sub-shot boundary detection. At well-defined moments, a frame is taken to give a quick overview of what is happening in a video, allowing e.g. quick navigation.

5.6 Motion recognition

The motion recognizer detects either motion of the camera (pan, tilt or zoom) or motion inside the scene (see [12] and [13]). This is particularly useful in case the user wants to distinguish between static or dynamic shots, or wants to know when and where a change in the background occurs. The results of the motion recognizer can also be used by other detector to compensate the effect of the camera motion while tracking objects or people inside of a scene. The detector processes about 25 frames per second.

5.7 Face detection

The face recognition detector, based on the Viola-Jones algorithm (see [14] and [15]), is used to identify the number of persons in a scene. The detector can be configured to find frontal faces, profile faces or both, and has also limited face tracking capabilities. The speed of the detector depends on the parameter set, but can reach 40 frames per second.

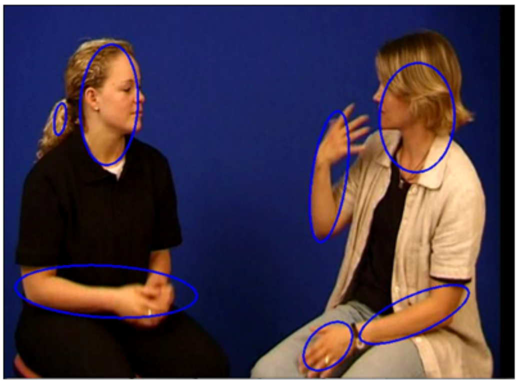


Figure 3. This figure shows the results of a detection of heads, hands and arms, based on a previous skin-color detection step.

5.8 Body part tracking

This detector identifies body parts (hands, arms and heads) and then tracks them. It estimates at first the skin colour (see [16] and [17]) for each shot in the video and then identifies and tracks the different body parts, which are then approximated by ellipses. By tracking the body parts the user knows when movements/signs begin or finish, when hands join, what is the position of the hands with respect to other body parts. This detector runs at anput 50 frames per second. An example of the results from this detector is shown in figure 3: Tracked body parts are marked with ellipses. Note that the detector does not yet have a body model, but tracks moving skin-color areas.

5.9 Gesture recognition

The gesture recognition tool identifies simple hand gestures, still or moving. This detector is still in early development and nor qualitative nor quantitative tests have been made. ELAN is already used for manual annotation of sign language (see [18] and [19]), but machine support could help to improve speed and quality of the annotation process a lot.

5.10 Robustness and user interface

Currently, we are testing the behaviour of the existing detectors with respect to the variety of material we have in our 800 GB test corpus (300 GB of audio and 500 GB of video, mostly WAV and MPEG 1, 2 and 4). It is obvious that we need to study, how we can create simple to use interfaces to allow users to influence detection parameters easily and to immediately see the effects. Moreover we would like to gather feedback from users in an iterative process to improve the quality of the analysis.

Using a common interface, detectors in AVATecH can be called either from ELAN or from a custom batch processing tool which we called ABAX (AVATecH Batch Executor). For that, each of the detectors comes with a metadata file which specifies the necessary parameters and input and output files to call that detector. While the metadata can define choice lists and the ranges for numerical parameters, it does not attempt to be a machine readable representation of the parameter semantics. Instead, it contains a short description of each item for use in human user interfaces which can be automatically generated from the metadata. Note that all parameters must have defaults: This helps the users to quickly get first results. Once they found a detector to be useful for their annotation work, they can adjust settings (for a group of input files or separately for each file) to improve the quality of the results.

Detectors can be made available for a number of operating systems, using a platform independent design for communication: Parameters, file names and log / progress information are sent through pipelines as plain (XML) text. This even allows the use of (intranet) “detector servers” by sending the pipelines through a TCP/IP connection. Caller and detector still have to share a (network) filesystem for media and (XML or CSV) result files. A direct Java API is also available, for cases where the focus is on tight integration.

6 SUMMARY

With integrating a number of detection components that create layers of annotations that can be easily used by ELAN and TROVA, we are making a new step in facilitating the work of manual annotators. Also, coarse automated annotations can help to find useful recordings in unexplored corpora. As has been seen from the very simple silence detector, which we used as first example, it can speed up the work of researchers by factors when the interaction interface is simple and the user can stay in a well-understood tool framework. A set of first low hanging fruit detectors has been tested and is being integrated into the ELAN framework. The results will be analyzed to determine which other more complex detectors will be added and how user interaction options need to be modified to maintain attractiveness for researchers who are not only interested in pure recognition scores but also want to understand underlying mechanisms.

ACKNOWLEDGEMENTS

AVATecH is a joint project of Max Planck and Fraunhofer, started in 2009 and funded by MPG and FhG.

REFERENCES

[1] L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy, ‘The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty’, *ACM Computing Surveys (CSUR)*, **12**(2), 213–253, (June 1980).

[2] K. Rohlfing, S. Duncan, et al., ‘Comparison of multimodal annotation tools - workshop report’, *Gesprächsforschung Online-Zeitschrift zur verbalen Interaktion*, **7**, (2006).

[3] S. Bird and M. Liberman. A formal framework for linguistic annotation (revised version), 26 Oct 2000.

[4] E. Auer, H. Sloetjes, and P. Wittenburg, *AVATecH Component Interface Specification Manual*, http://www.mpi.nl/research/research-projects/language-archiving-technology/avatech/, 2010.

[5] Shih-Sian Cheng, Hsin-Min Wang, and Hsin-Chia Fu, ‘Speaker segmentation using divide-and-conquer strategies with application to speaker diarization’, *IEEE transactions on audio, speech, and language processing*, **18**(1), (2010).

[6] K. Biatov and J. Köhler, ‘Improvement speaker clustering using global similarity features’, in *Proceedings of the Ninth International Conference on Spoken Language Processing*, (2006).

[7] K. Biatov and M. Larson, ‘Speaker clustering via bayesian information criterion using a global similarity constraint’, in *Proceedings of the Tenth International Conference SPEECH and COMPUTER*, (2005).

[8] D.A. Reynolds, ‘Speaker verification using adapted gaussian mixture models’, *Speech Communication Journal*, **17**(1-2), (1995).

[9] C. Petersohn, ‘Fraunhofer HHI at TRECVID 2004: Shot boundary detection system’, in *TREC Video Retrieval Evaluation Online Proceedings*. TRECVID, (2004).

[10] Petersohn C., ‘Sub-shots– basic units of video’, in *EURASIP Conference Focused on Speech and Image Processing, Multimedia Communications and Services*, Maribor, Slovenia, (2007).

[11] Boreczky J. and Rowe A., ‘Comparison of video shot boundary detection techniques’, *Journal of Electronic Imaging*, **5**(2), 122–128, (1996).

[12] N. Atzpadin, N. Kauff, and O. Schreer, ‘Stereo analysis by hybrid recursive matching for real-time immersive video conferencing’, *Trans. on Circuits and Systems for Video Technology, Special Issue on Immersive Telecommunications*, **14**(3), 321–334, (2004).

[13] Dumitras A. and B.G. Haskell, ‘A look-ahead method for pan and zoom detection in video sequences using block-based motion vectors in polar coordinates’, in *Proceedings of the 2004 International Symposium onCircuits and Systems*, volume 3, pp. 853–856, (2004).

[14] Viola P. and M. Jones, ‘Robust real-time object detection’, in *Second International Workshop on Statistical and Computational Theories of Vision*, (2001).

[15] M. Viola P., Jones, ‘Rapid object detection using a boosted cascade of simple features’, in *Conference on Computer Video and Pattern Recognition*, (2001).

[16] S. Askar, Y. Kondratyuk, K. Elazouzi, Kauff P., and O. Schreer, ‘Vision-based skin-colour segmentation of moving hands for real-time applications’, in *Proceedings of the 1st European Conference on Visual Media Production (CVMP 2004)*, London, United Kingdom, (2004).

[17] S. Masneri, O. Schreer, D. Schneider, S. Tschöpel, R. Bardeli, et al., ‘Towards semi-automatic annotation of video and audio corpora’, in *Seventh international conference on Language Resources and Evaluation (LREC)*, (2010).

[18] H. Lausberg and H. Sloetjes, ‘Coding gestural behavior with the NEUROGES-ELAN system’, *Behavior Research Methods, Instruments, Computers*, **41**(3), 841–849, (2009).

[19] O. Crasborn, H. Sloetjes, E. Auer, and P. Wittenburg, ‘Combining video and numeric data in the analysis of sign languages with the ELAN annotation software’, in *Proceedings of the 2nd Workshop on the Representation and Processing of Sign languages: Lexicographic matters and didactic scenarios*, ed., C. Vetoori, pp. 82–87, Paris, (2006). ELRA.