AVATecH – automated annotation through audio and video analysis

Przemyslaw Lenkiewicz[1], Binyam Gebrekidan Gebre[1], Oliver Schreer[2], Stefano Masneri[2], Daniel Schneider[3], Sebastian Tschöpel[3]

[1]Max Planck Institute for Psycholinguistics, Wundtlaan 1, 6525 XD Nijmegen, The Netherlands
[2]Fraunhofer-Heinrich Hertz Institute, Einsteinufer 37, 10587 Berlin, Germany
[3]Fraunhofer IAIS Institute, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{Przemek.Lenkiewicz, BinyamGebrekidan.Gebre}@mpi.nl
{Oliver.Schreer, Stefano.Masneri}@hhi.fraunhofer.de
{Daniel.Schneider, Sebastian.Tschoepel}@iais.fraunhofer.de

Abstract

In different fields of the humanities annotations of multimodal resources are a necessary component of the research workflow. Examples include linguistics, psychology, anthropology, etc. However, creation of those annotations is a very laborious task, which can take 50 to 100 times the length of the annotated media, or more. This can be significantly improved by applying innovative audio and video processing algorithms, which analyze the recordings and provide automated annotations. This is the aim of the AVATecH project, which is a collaboration of the Max Planck Institute for Psycholinguistics (MPI) and the Fraunhofer institutes HHI and IAIS. In this paper we present a set of results of automated annotation together with an evaluation of their quality.

Keywords: Annotations, multimodal analysis, signal processing

1. Introduction

In recent decades we can experience a tremendous amount of changes in Languages and cultures. UNESCO has reported that currently one language becomes extinct every two weeks and even major languages are changing. During the last decades we recognize an increasing awareness about these threats resulting in a number of world-wide initiatives to document, archive and revitalize languages (DOBES, HRELP, PARADISEC). It is well understood now, that we have the obligation to preserve our material and knowledge about languages for future generations, since they may want to understand their roots. During the last decade also the awareness has grown that making recordings alone is not sufficient to guarantee that future generations will indeed be able to access the data. Recordings without appropriate annotations and metadata can be completely useless for anybody that has no knowledge about their creation and purpose. Therefore a significant role in the archiving tools will be played by the automated annotation algorithms, which are developed as part of the AVATecH project (Wittenburg et al., 2010). The project aims at designing and implementing algorithms that allow for the automatic and semi-automatic creation of pre-annotations for the corpora, hence reducing the time needed to perform the manual annotation task. Their role is twofold: 1) they would allow a decrease of time necessary to perform this task, which is normally very laborious; 2) automation of some parts of the process can greatly increase the uniformity of the annotations created worldwide by different researchers, which would contribute to consistency of the available language data. In this paper we describe in detail the algorithms that operate on video recordings and present the initial results that we could obtain with them.

This task if a very challenging task due to two factors: 1) the size of the media corpora is very significant, reaching 70 TB presently; 2) the recordings are of very high diversity of languages, conditions and situations. This means that effective methods for automated processing of such content are not widely available or don't exist at all.

2. Audio and Video Analysis Algorithms

The main principle that led the development of video analysis algorithm was to reduce the time needed to perform the annotation process and, when possible, make it completely automatic. The creation of robust and efficient algorithms was mandatory, due to the huge size of the video database of the MPI and the great diversity of the content. These two constraints were the main guideline in the creation of new algorithms and in the adaptation of existing ones to this specific problem. All the algorithms are designed to work without user interaction, except for the initial setup of some parameters. This approach was chosen because of the fact that potential users can have a limited technical knowledge and also to save as much time for the researchers as possible.

The implementation is done using a highly modular structure, so basic functionality delivered by some recognizers can be used as building blocks to solve more complex tasks.

2.1 Audio Segmentation

For linguistic annotation, segmentation on the utterance level is of high importance, but hard to achieve automatically without errors. This recognizer provides a fine-granular segmentation of the audio stream (Cheng 2010) into homogeneous segments, e.g. between speakers or at other significant acoustic changes. The user can control the granularity of segmentation by tuning a corresponding feedback parameter.

2.2 Speech detection

This recognizer is able to label audio segments

containing human speech, regardless of the language of the recording. The user is allowed to manually provide a small amount of speech and non-speech samples in order to adapt the model to the given data, which leads to a more robust detection.

2.3 Speaker clustering

A language-independent speaker clustering recognizer is able to find segments spoken by the same person within a given recording. (Biatov and Kohler 2006; Biatov and Larson 2005; Reynolds 1995). 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.

2.4 Vowel and pitch contour detection

The pitch contour detector can allow researchers to graphically specify pitch contours and search for similar patterns. The detector can tag segments in audio recordings and annotate 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.

2.5 Shot/cut detector and keyframes extractor

Shots and sub-shots are defined as a sequence of consecutive frames showing one event or part thereof taken by a single camera act in one setting without change in visual content, or with a very small change. Our algorithms process standard definition videos at about 130 frames per second, on a Pc with Intel Xeon, 2.53GHz.

The number of videos in the database is so big and increasing at such a fast pace that often the researchers don't even have the chance to watch the video to decide whether it is worth annotating it. That's why one of the first requests from the linguist was to realize a tool that, even if it doesn't help in the creation of new annotations, allows them to browse easily and quickly the content of a video. The key frames extraction tool takes as input the information provided by the shot/cut detector and extracts an image each time a sub-shot is detected. Using a standard configuration the processing speed is 5 to 10 times faster than real-time.

2.6 Global motion detection

Another useful feature that can provide useful information to the researchers is the detection of motion in a video, which can allow distinguishing between different types of video content. E.g., the presence of zooms and motion inside of a scene are usually the most interesting, while shots containing just panning and a low amount of internal motion are usually of little interest and can be usually discarded without further analysis.

For each frame in the video a motion vector map is computed using the Hybrid Recursive Matching (HRM) algorithm (Atzpadin et al. 2004). The approach used for zoom detection is similar to (Dumitraş and Haskell, 2004) and is based on the idea that when a zoom happens the majority of motion vectors point to (or come from) the center of the frame

2.7 Skin color estimation

This skin color estimator does not need a training dataset but rather estimates the YUV ranges identifying skin color for each frame in each video. The algorithm uses both the temporal information provided by the change between one frame and the next and the spatial information provided by the fact that skin color pixels tend to cluster in well defined regions. It works in two steps: at first it uses a change detection tool to select the most suitable frames for skin color estimation, and then it applies an iterative clustering algorithm to select the range in the YUV domain that best represents skin color.

The idea of the change detection step is to apply a change detection algorithm to the luminance component of consecutive frames of the video and to obtain then a binary image (the change image) that is set to one for pixels where the difference in value between the frames is above a certain threshold. A 2D histogram of the change image is then computed and its bins are grouped into clusters. Ideally, each one of these clusters represents a body part moving in the current frame. Information regarding size, position, compactness is recorded for each cluster found in the histogram. After that all this information is passed to a cost function, which assigns a score to the current frame based on the properties of the clusters. The higher the score, the higher the probability that arms and heads are not overlaying, making the subsequent skin color estimation possible. The three frames obtaining the highest score are then selected to perform the second step of the algorithm, the iterative skin color estimation.

At this step the algorithm segments the selected frames, marking the pixels in the image if they are within a specific range in both the U and V components. In this way for each UV interval under consideration a corresponding binary image is obtained and, a cluster analysis is performed to decide which range is the most likely ones to represent human skin color. The decision of the best color range is based on the number of clusters retrieved, their size, their compactness (defined as the ratio between the number of segmented pixels and the area of the ellipse that best approximates the shape of the segmented skin region) and their position with respect to the position of the clusters found analyzing the change image.

2.8 Hands and heads tracking

The algorithm works at first by segmenting the image in skin vs. non-skin pixels, using the information provided by the skin color estimator. The subsequent step in the detection process involves the search of seed points where the hands and heads regions most likely occur. Histograms along the horizontal and vertical directions compute the number of pixels with luminance and color values within the desired interval; the pixels where a maximum occur in both the directions are selected as seed points. A region-growing algorithm is then applied to the seed points in order to cluster together all the skin pixels in the neighbourhood. Each region is approximated by an ellipse, characterized by the position of the center, its orientation and the length of its axes and for tracking purposes each of them is assigned a label. The tracking is performed by analyzing the change in position and orientation of the ellipses along the timeline, assigning labels based on position of the regions in the current and previous frames.

2.9 User interaction

The expected data is very heterogeneous and in some cases baseline recognizers can perform poor with no additional adaptation. Furthermore the researchers cannot accept annotation errors, e.g., a segment that is wrongly labeled as no-speech but has speech in it (false negative). Therefore the analysis components support adaptation and feedback-loop mechanisms. By adaptation mechanism we mean that the researcher is able to give examples of aspects he likes to detect, e.g., samples of a speaker for automatic speaker detection or sample segments without speech for the automatic detection of speech. By feedback-loop mechanism we mean strategies where the user runs a recognition process at first, then gives feedback about the quality of the result and then runs the process with the updated information again. For example, this could be applied for the speaker identification process: The user adapts the recognizer before running the component the first time

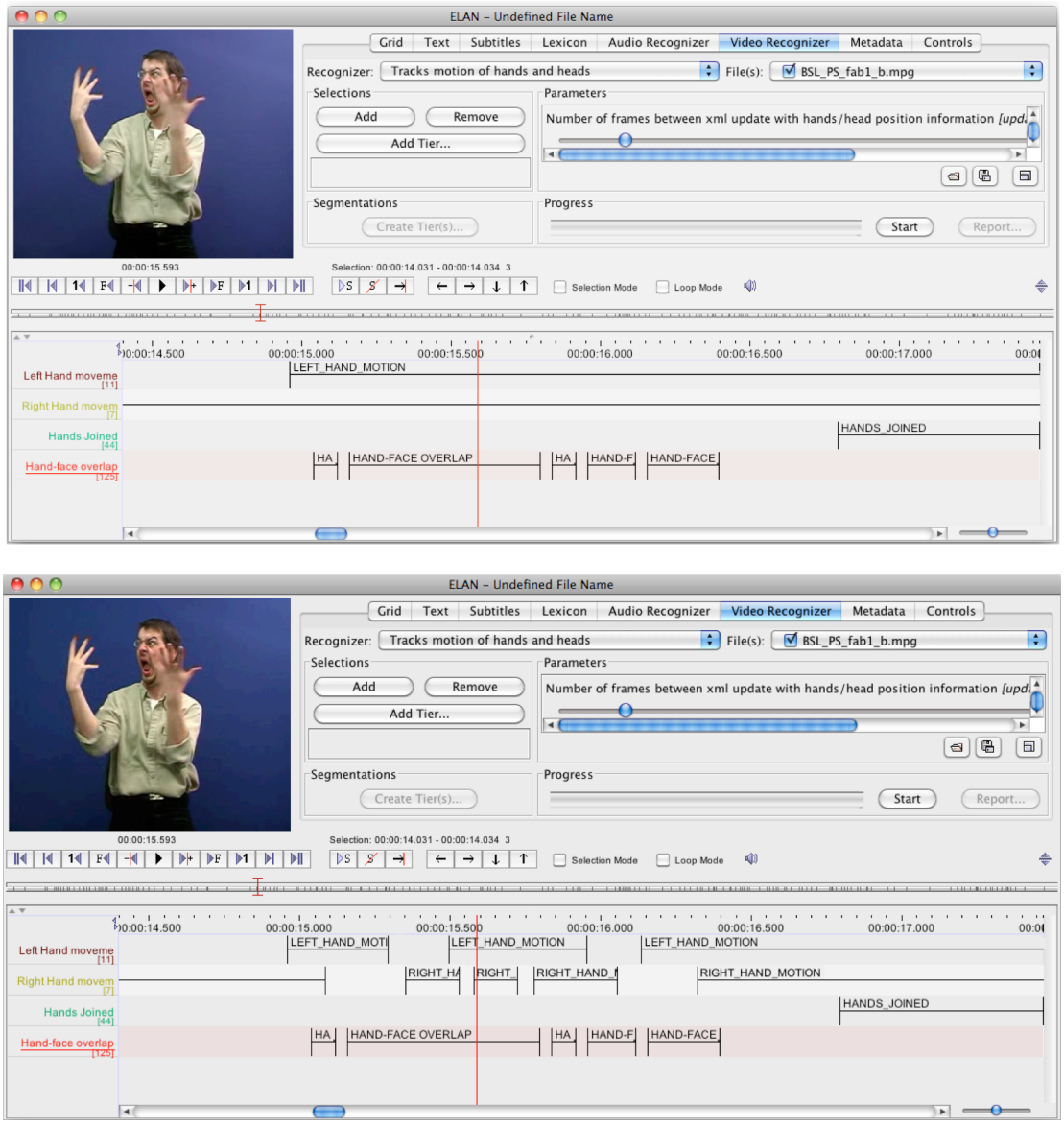

Figure 1. Comparison of the stroke detection recognizer executed with high (top image) and low (bottom image) thresholds.

by selecting some examples of the speaker, then runs the recognizer, and then verifies a number of segments and the recognizer would use this response to adapt the algorithm before running the process again.

3. Results and Experiments

To test the developed methods we have prepared a range of tasks, to be solved by a researcher with the help of the recognizers. The time necessary to complete these tasks will be compared with the time that takes to perform such task manually, without the support of recognizers. The said tasks have been defined to take advantage of audio and video recognizers.

We have executed the video and audio recognizers with exemplary files, taken from Max Planck Institute corpus. The resulting annotations are presented on Figure 1. It presents automated segmentation of video recording of a person telling a story in sign language. The gestures of the said person have been analyzed and individual strokes have been detected. The showed example presents 4 features that have been discovered, namely the strokes performed by both left and right hands, joining of both hands together and overlapping of the face by any of the hands.

The said features have been detected successfully and with high precision. The process was performed with very little user interaction and has consisted of the following steps:

- The recording has been segmented into consecutive shots, given by camera operation or other significant changes in the scene. No user interaction was required at this point.
- Color values representing human skin have been estimated. This process is performed automatically under the assumption that the objects that move the most in analyzed video are human hands and/or head. The result is given for user verification and it is possible to correct it quickly using a functional application with a graphical user interface (Figure 2). Movement of the sliders is immediately visible in the preview window, which makes the process very intuitive.
- The positions of hands and head are detected for every frame of the video and the coordinates of those are recorded. No user interaction is required.
- Using the coordinates of hands and head and the time information (directly from the video) the gestures and strokes are detected. The required user interaction is limited to choosing a value of the threshold, which will be used to evaluate if the given movement of the hand can be classified as a stroke. An example of an experiment with a low threshold can be seen on Figure 1 bottom. As it is possible to see, smaller strokes are detected and when the hand goes into rest for a short time, a new stroke is considered. This precision allows detecting very subtle gestures and labeling them accordingly. Furthermore, the possibility to define this threshold allows different researchers to adapt the recognizer to their needs and find either longer or shorter strokes.

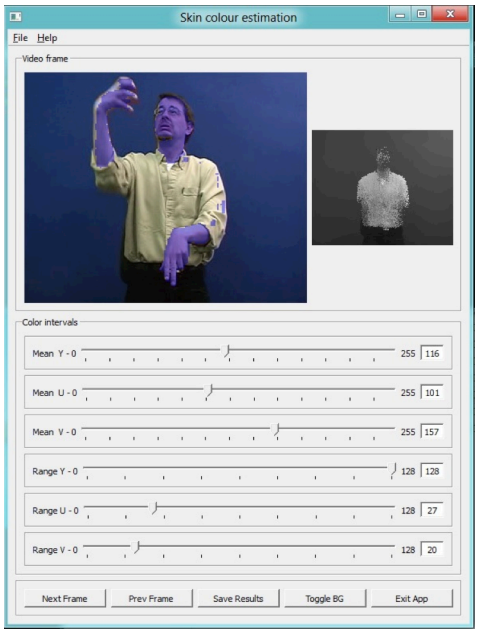

Figure 2: Graphical user interface application that allows to adjust the color ranges representing human skin. The result is marked purple, so the user can easily see how good is the selection. The small image on the right represents what is considered to be background. This allows further improving the detection of hands and head in the video.

The same recording has been presented for researchers of the MPI in order to perform manual segmentation of the same kind. The results of our experiments are highly consistent with the results obtained by manual annotation. Researchers participating in the experiment have corrected the results of the automated annotation in a way that it corresponds to their need. Each time the needed only a fraction of time, which they earlier spent on manual annotation. The average time which each of them has taken to correct the automated annotation equals 0.23 of the average time spent by them on manual annotation.

The second experiment consisted of marking the utterances of every speaker in the recording. This tasks if often performed as one of the first steps during any recording annotation. No high expertise is required, but nevertheless it takes very significant amounts of time. Again our algorithms have been executed with very little user interaction and according to the following plan:

- Audio signal has been segmented into homogenous parts using the standard audio segmentation recognizer. No user interaction was necessary.
- The parts that hold human speech have been labeled using the speech detection recognizer.

No input from user was necessary. However, if the results would prove not satisfactory, the user has the possibility of providing examples of speech and non-speech segments in the recording. This ways the user is capable of incorporating their own knowledge in the process and improve the classification results.

- Speakers in the recordings have been identified and proper identifiers have been assigned to each segment. No user interaction is necessary at this step.

The results showing the automated annotation can be seen on Figure 3. Similarly to the previous case the automated results have been given to MPI researchers and they have adjusted them to their needs. The necessary work included correcting the boundaries of the segments, fixing the overlapping of speakers (this is not detected by the recognizers) and joining several separate speakers as one person (the opposite situation, detecting two persons as the same speaker, did not occur in our tests). Again, the time necessary to perform these corrections has been significantly shorter than in case of doing the annotation manually and was on the average 0.38 of the average time spent on manual annotation.

4. Performance

The time necessary to execute the recognizers on the recordings has not been taken into account in our experiments, as it is considered to be of low importance because of the very little user involvement. After setting the initial parameters the researchers can carry out their work with different tasks. Also their computer is not used for the heavy computations, our project assumes that the algorithms are executed on local network servers, which store all the video and audio data of MPI researchers, and only the resulting annotation files are returned to users' computers. The waiting time is also not very long as all the recognizers perform between 2-10 times faster than real time, which means that a minute of a recording will take between 6 and 30 seconds to be processed by any recognizer.

5. Conclusions

The specification and implementation of the above-described video processing recognizers has been performed in a very close contact with linguist researchers and according to the needs they have specified. After testing the relative effectiveness of our methods and witnessing the dramatic decrease of time necessary for annotations, we can say that our goals have been chosen correctly and our methods have proven very useful. As our next steps we are planning to fully develop the possibility of detecting and tracking the hands in the videos, differentiate left from right one and also work together with linguists to develop new recognizers that would create new types of annotations, for different research questions. We believe this work would contribute significantly to the quality of linguistic data stored at the Language Archive of MPI and possibly in other locations.

6. Acknowledgments

AVATecH is a joint project of Max Planck and Fraunhofer, started in 2009 and funded by both Max Planck Society and Fraunhofer Society.
Some of the research leading to these results has received funding from the European Commission's 7th

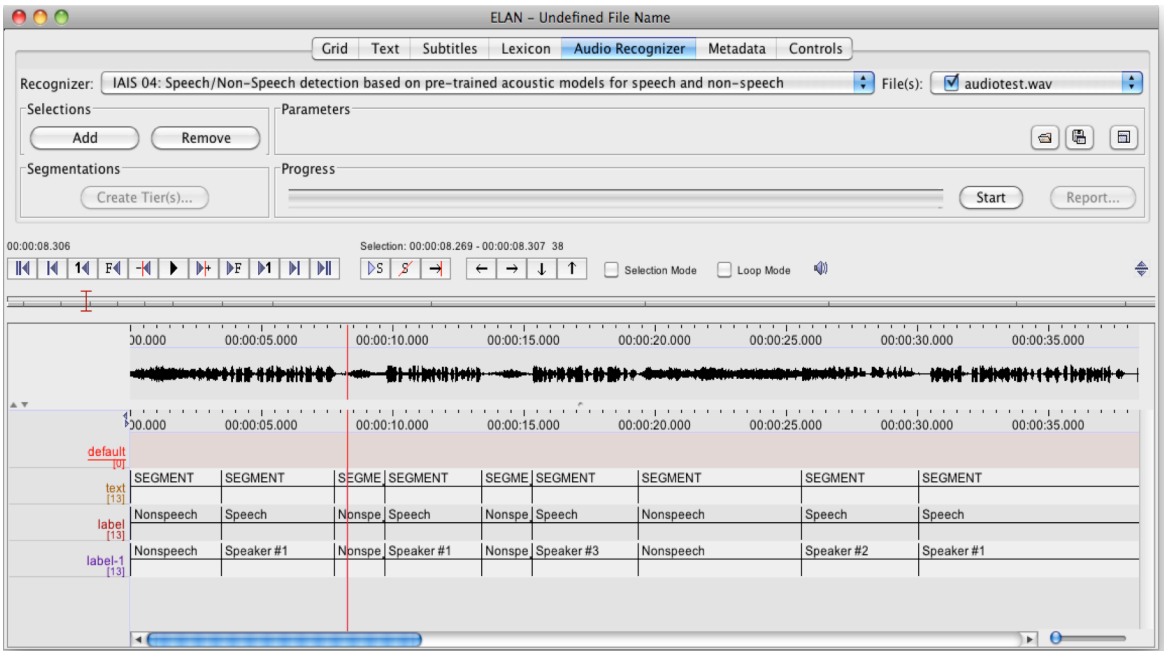

Figure 3. Three layers of annotation representing three steps of the automated analysis. First layer is the result of the uniform document segmentation, second layer is the division into speech/no-speech parts, third layer recognizes different speakers in the recording. All the steps have been performed automatically.

Framework Program under grant agreement n° 238405 (CLARA).

7. References

Atzpadin, N., Kauff, P., Schreer, O., (2004), *Stereo Analysis by Hybrid Recursive Matching for Real-Time Immersive Video Conferencing*, Transactions on Circuits and Systems for Video Technology, Special Issue on Immersive Telecommunications, Vol.14, No.3, 2004, pp. 321-334.

Biatov, K., and Kohler, J., (2006), *Improvement speaker clustering using global similarity features*, in Proceedings of the Ninth International Confer- ence on Spoken Language Processing.

Biatov, K., and Larson, M., (2005), *Speaker clustering via bayesian informa- tion criterion using a global similarity constraint*, in Proceedings of the Tenth International Conference SPEECH and COMPUTER.

Dumitraş, A., Haskell, B.G. (2004). *A look ahead method for pan and zoom detection in video sequences using block-based motion vectors in polar coordinates.* Proceedings of International Symposium on Circuits and systems, ISCAS 2004, Vancouver, Canada, May 2004.

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

Wittenburg, P., Auer, E., Sloetjes, H., Schreer, O., Masneri, S., Schneider, D., Tschopel, S., (2010), *Automatic annotation of media field recordings,* 4th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, LaTeCH, Lisbon, Portugal.