Joke-o-mat: Browsing Sitcoms Punchline by Punchline

Gerald Friedland
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704-1198
fractor@icsi.berkeley.edu

Luke Gottlieb
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704-1198
luke@icsi.berkeley.edu

Adam Janin
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704-1198
janin@icsi.berkeley.edu

ABSTRACT

This paper summarizes our contribution to the Yahoo! task of the ACM Multimedia Grand Challenge. This challenge asks for the robust automatic segmentation of videos according to "narrative themes". Based on the automatic segmentation methods presented in [1] and partly [2], we describe a system to navigate Seinfeld episodes based on automatic segmentation of the audio track only. The system distinguishes laughter, music, and other noise as well as speech segments. Speech segments are identified against pre-trained speaker models of the actors. Given this segmentation and the artistic production rules that underlie the genre *situation comedy* and Seinfeld in particular, the system enables a user to browse an episode by scene, by punchline, and by dialog segments. The themes can be filtered by the main actors, e.g. the user can select to see only punchlines by Jerry and Kramer. Based on the length of the laughter, the top 5 punchlines are also identified and presented to the user.

Categories and Subject Descriptors

H5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Signal analysis, synthesis, and processing*; H5.4 [**Information Systems Applications**]: Navigation

General Terms

semantic segmentation

Keywords

acoustic event detection, speaker id, video navigation

1. INTRODUCTION

The following article briefly presents our contribution to the Yahoo! challenge of ACM's Multimedia Grand Challenge[1]. Based on the idea that TV shows are already segmented into narrative themes, such as scenes and dialog acts, by their producers, we present a system that analyzes these "markers". We chose to stick to an example presented in the challenge description, namely the segmentation of Seinfeld episodes. Sitcoms, being invented as a radio-format in the 1920s, typically follow a strict set of rules encoded in the audio, e.g. every scene transition is marked by a piece of music and every punchline is labeled by artificial laughter. Although integrating with video cues would likely improve the system in various aspects, we chose to limit ourselves to audio both to show how far one can get using only audio analysis and because of resource and time constraints.

We assume the following use case: The first time a person watches a Seinfeld episode, he or she needs barely any navigation. Unlike other media, such as recorded lectures, sitcoms are designed for entertainment and should hold the attention of the viewer for the entire length of an episode. A play and a pause button should be sufficient. Any involuntary interruption of the flow of the episode might detract from the experience. When a sitcom is watched at later times, however, a user might want to show a very funny scene to a friend, point out and post the sharpest punchline to his or her facebook network, or even create a home-made YouTube video composed of the most hilarious moments of his or her favorite actor. In order to do this quickly, a navigation interface should support random seek into a video. Although this feature alone makes search for a particular moment in the episode possible, it remains cumbersome, especially because most sitcoms don't follow a single thread of narration. Therefore, we present the user with the basic narrative elements of a sitcom such as the scenes, punchlines, and individual dialog segments on top of a standard video player interface. A per-actor filter helps to search only for elements that contain a certain protagonist. The user is now able to selectively skip certain parts and to directly navigate into elements he or she remembers from the past. While keyword search based on speech recognition would improve the search capabilities, the method we present rather allows people to *surf* the episode.

2. TECHNICAL APPROACH

The system consists of two main elements: First, a preprocessing and analysis step and second an online video browser. The preprocessing step consists of an acoustic event detection and speaker identification step and a narrative element segmenting step. The online video browser then uses the output of the narrative elements analysis step to present a video navigation interface to the user.

[1]For a demo see `www.icsi.berkeley.edu/~fractor/mmch/`

MM'09, October 19–24, 2009, Beijing, China

2.1 Acoustic Event and Speaker Identification

For both acoustic event detection and speaker identification, we use a derivate of the ICSI speaker diarization engine [2] used for speaker identification in meetings [1]. Similar to our experiments with meetings, the speech is single-channel. However, the speech data contained in sitcoms have slightly different properties compared to speech recorded in meetings. TV shows are usually recorded using a boom microphone with much better signal quality than for meetings. Most importantly, though, unlike real-world meetings there is very little overlapped speech in sitcoms. So in general, detecting acoustic events and speakers in a TV show is easier than in a real-world meeting recording.

One disadvantage over meeting recordings is that speech is very often overlapped with either music or artificial laughter. While it is often possible to separate out the music/laughter track by implementing a 4-on-2 surround decoder because laughter/music and speech are usually on a different channels, we found that there is little benefit in doing so as the methods presented in [1] seem to be quite robust against overlapping laughter and music.

Similar to [1], we trained 60 seconds of each of the following classes: Jerry, Kramer, Elaine, and George, male supporting actor, female supporting actor, laughter alone, music alone, and other noise. The speakers were trained with both pure speech and laughter and music-overlapped speech. The audio track of the episode is then segmented in 2.5 s chunks and each of the chunk is classified to belong to any of the above mentioned classes. The result is saved in NIST-standardized rttm-format (see [2]).

In order to evaluate the experiments, we hand annotated the Seinfeld episode "The Soup NAZI" completely. We did not use forced alignment and we only used one annotator. Therefore performance comparison to [1] or [2] is difficult. We obtained a Diarization Error Rate of 46 %, which is an error of about 5 % per class and therefore roughly consistent with [1]. We expect the system to perform much better with more training data and the exact annotation of all supporting actors might improve the system. We made two different demos available: One that is based on the acoustic event detection and speaker identification and one the human annotated data. The automatic narrative theme analysis is de-facto perfect when performed on the manually generated speaker, music, and laughter labels.

2.2 Narrative Theme Analysis

The narrative theme analyzer is a rule-based system that transforms the segmentation generated by the acoustic event and speaker detection into segments that reflect narrative themes. We expect having to adjust these rules for different TV series, e.g. "I love Lucy" might have similar but not identical rules. The rules for the Seinfeld themes are as follows: A *dialog element* is a single contiguous speech segment by one speaker. A *punchline* is a dialog act that is followed by pure laughter. Punchlines are prioritized by the length of this laughter segment. The longer the laughter, the more important is the punchline. The *top-5 punchlines* are the 5 punchlines followed by longest laughter. A *scene* is a segment of at least 10 seconds between two music events or a music event and the beginning or end of file.

Once models are trained, the complete preprocessing of a video takes about 0.5 × realtime on a current standard PC. Training time is about 0.3 × realtime per model.

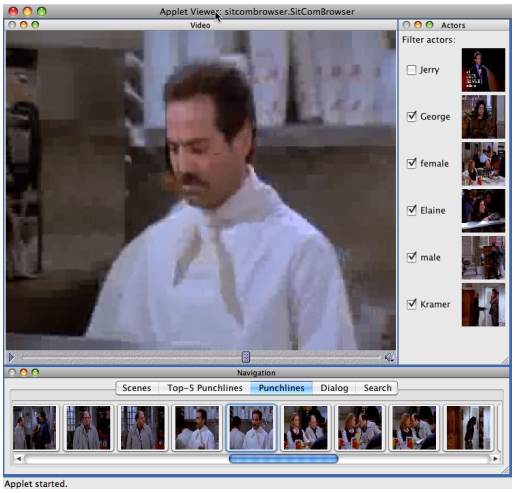

Figure 1: The narrative theme video browser.

2.3 Video Browser

We imagine the browser (see Figure 1) to replace a typical YouTube video player. The browser shows the video and allows play and pause, as well as seeking to random positions. The navigation panel on the bottom shows iconized frames of the video. The frames are grabbed at 50 % duration of the narrative element that it represents. The navigation panel allows the user to directly jump to the beginning time of either the scene, punchline, top-5 punchline, or dialog element. Also, the current narrative element is highlighted while the show is playing. In order to make navigation more selective, the user can deselect one of the main actors or the male/female supporting actors. In this case, scenes, punchlines, or dialogs that only contains deselected actors are no longer visible in the navigation bar. The actor icons are currently grabbed from the center of their longest dialog segment; we imagine using the localization approach presented in [2] to obtain better results in the future.

3. CONCLUSION AND FUTURE WORK

This article presents our approach to segmenting and browsing sitcoms according to narrative themes. The main idea is to exploit the artistic production rules of the genre, which specify how narrative themes should be presented to the audience. Since sitcoms were invented as a radio format, narrative themes are strongly marked in the audiotrack which allows for a very efficient analysis. We believe the automatic segmentation could still be drastically improved by training models more accurately and with more data. Also, combining the methods presented in this article with visual analysis (e.g. face detection for the actor icons) will undoubtedly improve the user experience. Therefore a multimodal approach should be the next step.

4. REFERENCES

[1] G. Friedland and O. Vinyals. Live speaker identification in conversations. In *Proceedings of ACM Multimedia*, pages 1017–1018. ACM, October 2008.

[2] G. Friedland, C. Yeo, and H. Hung. Visual speaker localization aided by acoustic models. In *Proceedings of ACM Multimedia*, to appear. ACM, October 2009.