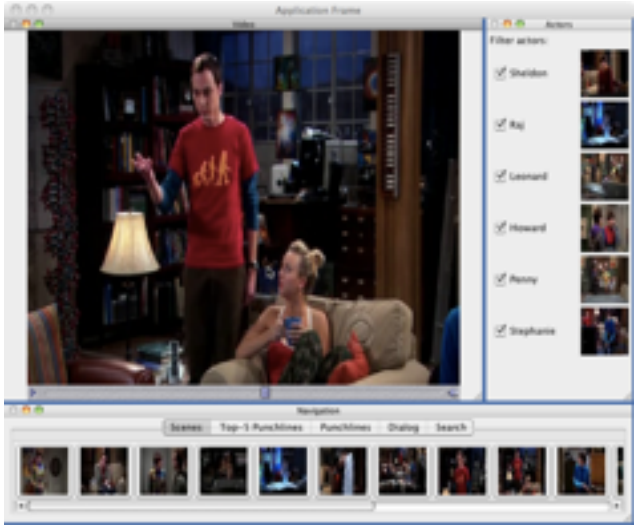
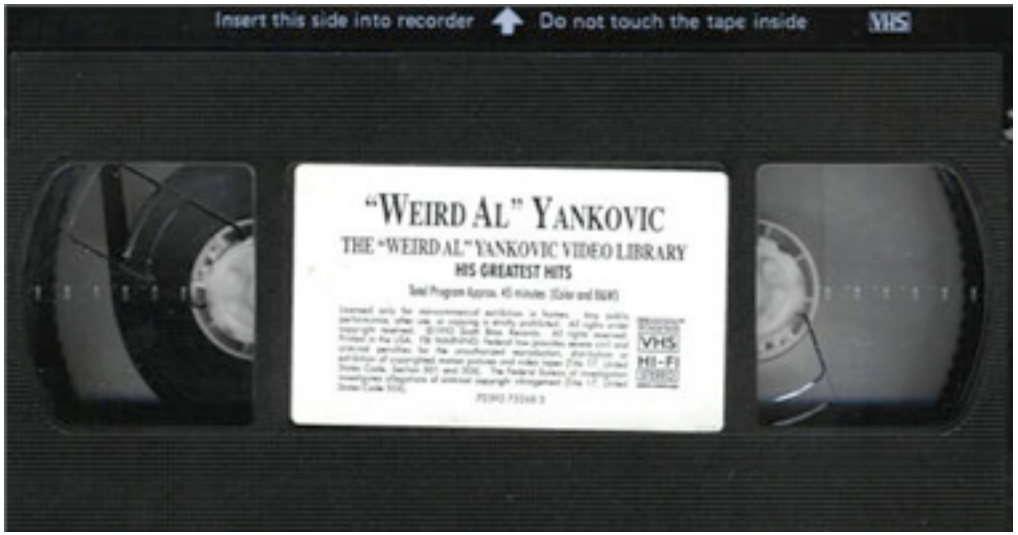
Narrative-Theme Navigation for Sitcoms Supported by Fan-Generated Scripts



Gerald Friedland, Luke Gottlieb, Adam Janin
International Computer Science Institute
Berkeley, CA
fractor@icsi.berkeley.edu

Motivation

Navigation Capabilities: 1985

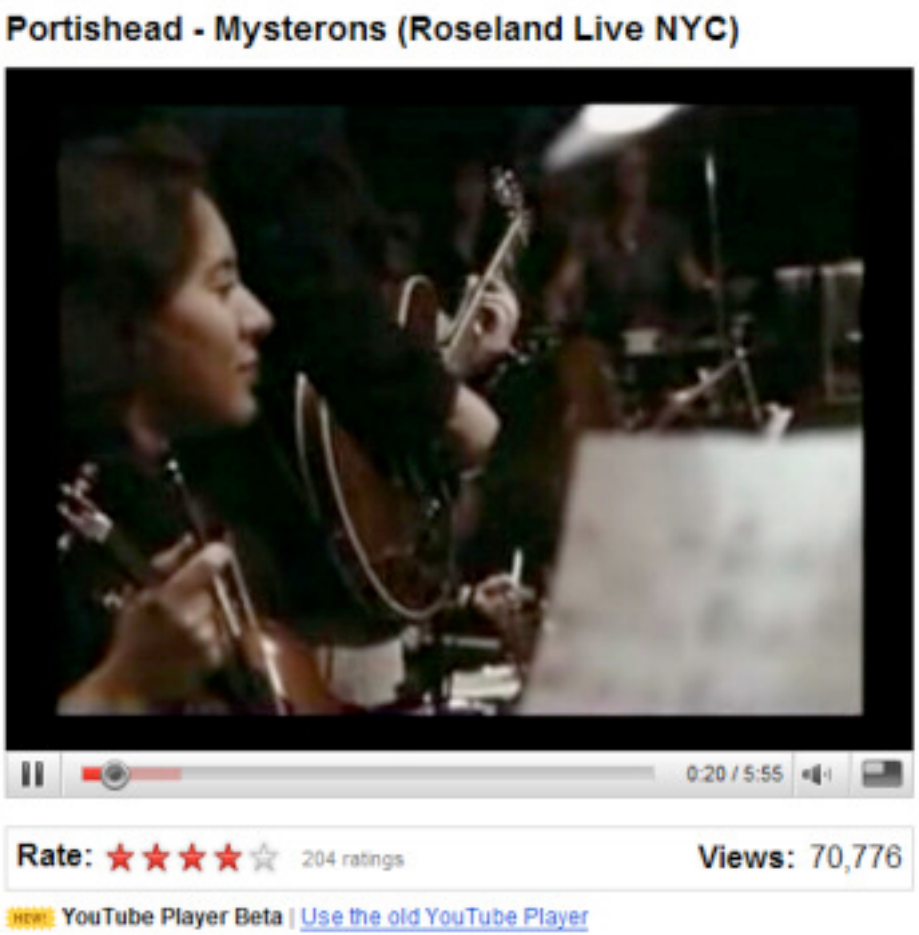


- Record
- Store
- Play
- Random Seek
- Annotate Manually

Motivation

Navigation Capabilities: 2010

- Record
- Store
- Stream
- Play
- Random Seek
- Annotate Manually



Motivation

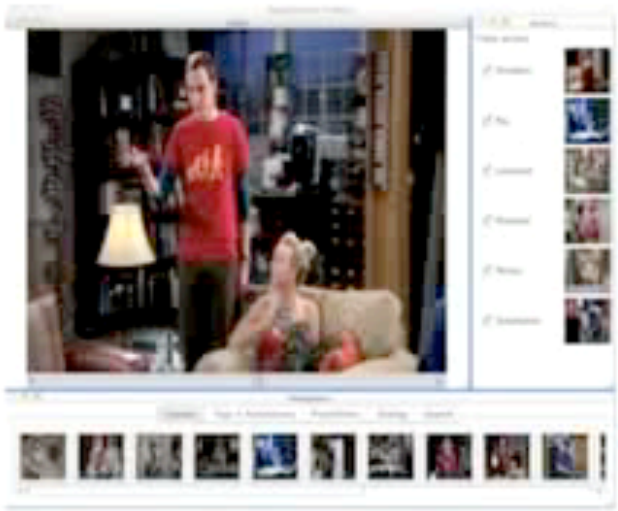
Yahoo! Video Segmentation Task of ACM Multimedia Grand Challenge 2009:

“Create navigation based on narrative elements”

INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Joke-o-Mat v1.0

Joke-O-Mat: Browsing Sitcoms Punchline by Punchline



Gerald Friedland, Luke Gottlieb, and Adam Janin

[friedland|gottlieb|janin]@icsi.berkeley.edu

Idea

TV content already has strict formatting.

Audio markers in many sitcoms:

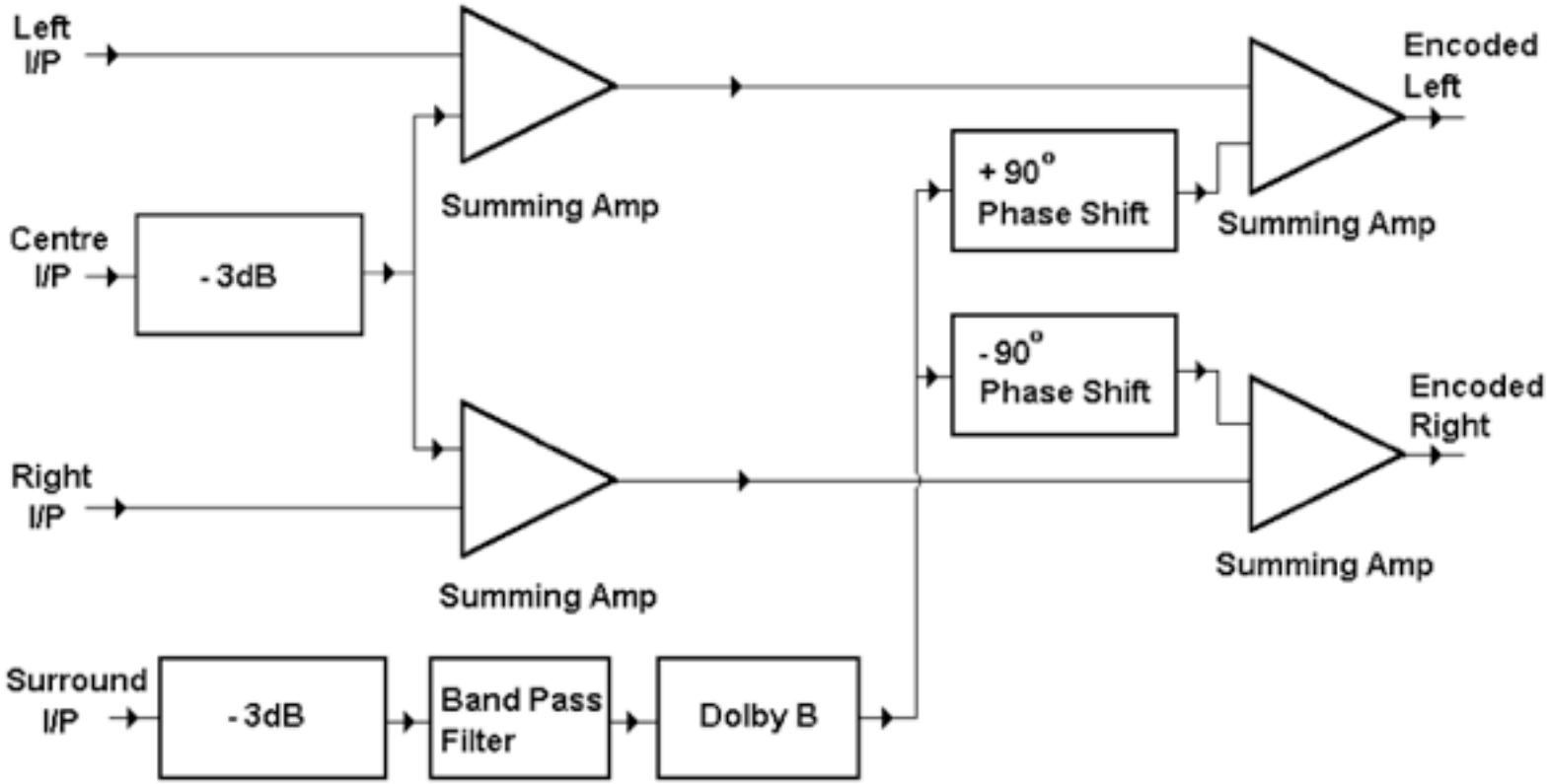
- Music: Scene change
- Laughter, applause: Punchline
  - Duration gives ranking of punchlines
- Voices of the actors
- Dolby Surround

Idea

TV content already has strict formatting.

Property	I Love Lucy	Seinfeld	Big Bang Theory
Air-Date	1952	1995	2008
Length	25:19m	21:51m	19:40m
Speech pauses	10.7%	10.5%	11.4%
Speech time 1st actor	19.8%	19.4%	18.2%
# Speakers >3% time	6	6	6
Speaker turns/minute	12	11	12
Overlap	2.1%	1.8%	0.5%

Dolby Surround

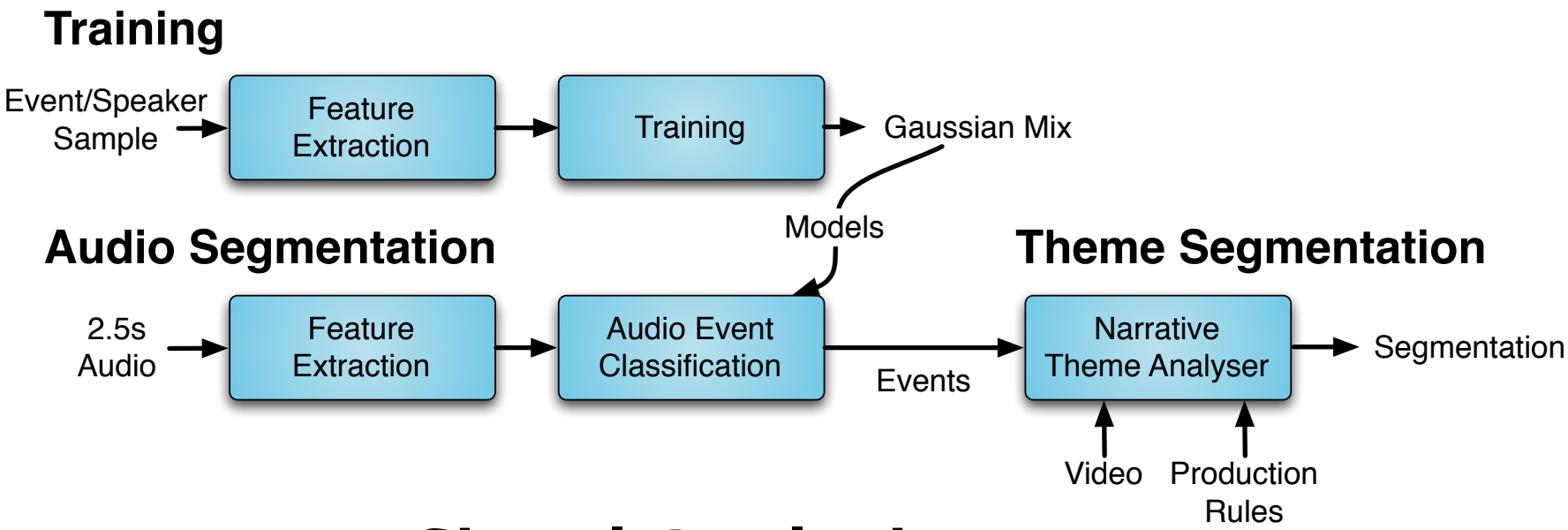


Implementation

Narrative Themes:

- Dialog element = contiguos speech
- Punchline = speech followed by laughter or applause
- Top-n punchline = punchlines followed by longest laughter, applause
- Scene = segment inbetween scene transition music

Implementation



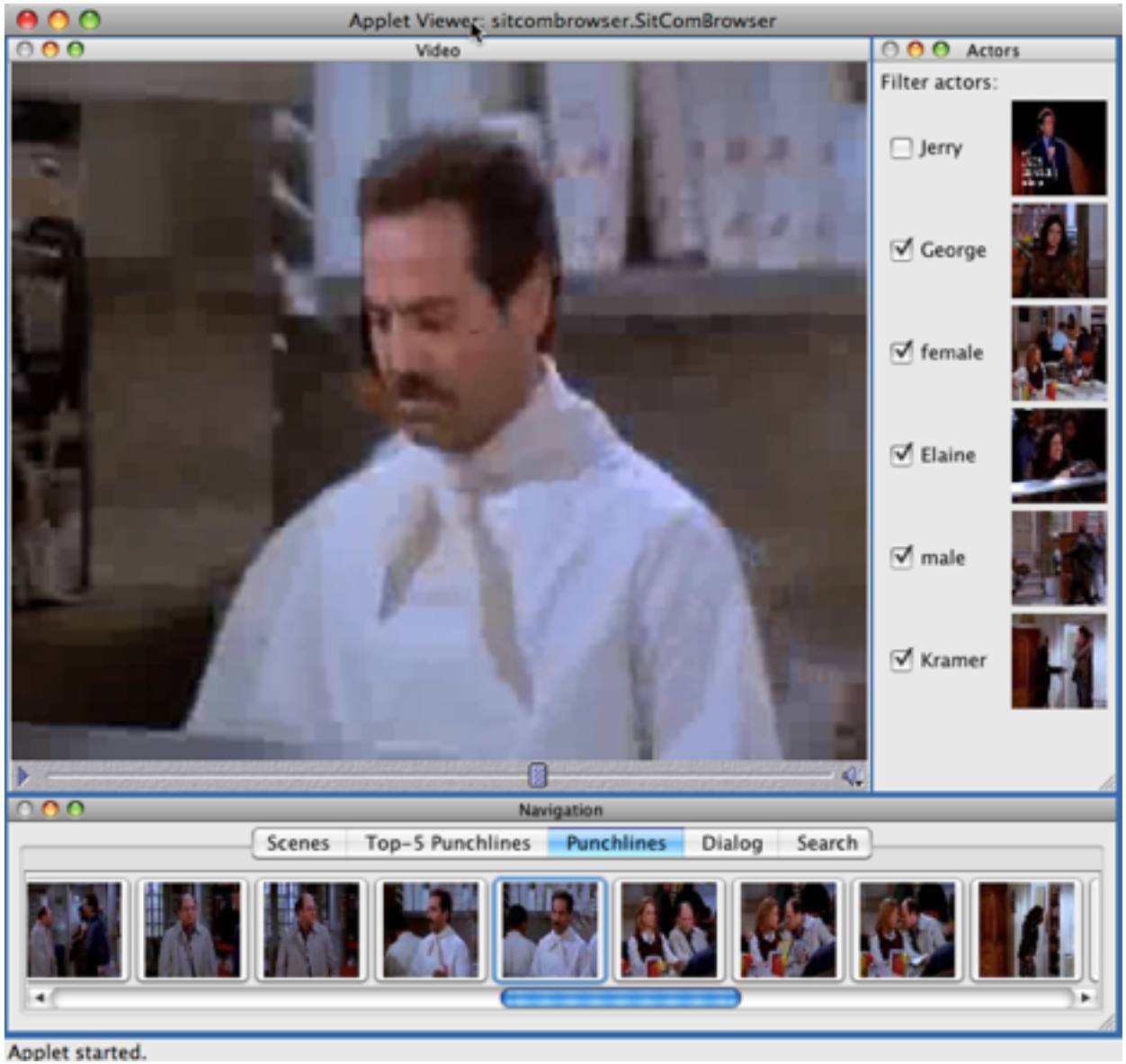
Signal Analysis:

- Acoustic Event Detection
- Speaker Identification

Objective Evaluation

Sitcom	DER	Sp/Nsp
Seinfeld	23.25%	0.9%
Seinfeld (Dolby trick)	47.12%	0.3%
I Love Lucy	32.04%	0.06%
Big Bang Theory	41.70%	0.5%

Interface


Applet Viewer: sitcombrowser.SitComBrowser
Video
Actors
Filter actors:
Jerry
George
female
Elaine
male
Kramer
Navigation
Scenes
Top-5 Punchlines
Punchlines
Dialog
Search
Applet started.

Subjective Evaluation

Turing-test Evaluation

- 10 “experts” from Appscio and Credit.com
- 6 preferred automatic version (!)

Limits of the Approach

- Training data required (per sitcom, per episode)
- Manual labeling of actors names
- Sometimes not even enough training data for supporting actors
- Genre/Format-specific ontologies: What if there's no laugh track?
- Doesn't take into account what was said

First Idea: Use Speech Recognition

Sitcom	WER
Seinfeld: Soupnazi	~80%

Using state-of-the-art ASR.

=> What about a different solution...

Joke-o-Mat v2.0

Idea

Get help with context:

- Close Captions (OCR)
- (Fan-generated) Scripts

SEINFELD

"OUT OF THE PAPER BAG"

Place: New York Airport

Who: Jerry, Elaine, George

Scene: Standing in line to board airplane to Boston.

George: Obsessing about getting on the plane and getting a good seat. Holding #111

Elaine: Carrying a paper shopping bag with handles.

Jerry: What's in the bag?
Elaine: Nothing
Jerry: Whatta ya mean "nothing"
Elaine: Nothing, Nothings in the bag!!
Jerry: Then why are you carrying it?
Elaine: I just like carrying something.
Jerry: An empty bag??

Example: Closed Caption

```
10.01 -> 13.21 -- WHERE' WE GOING?
                  TO THE RESTAURANT?
                  WHATEVER
14.72 -> 15.22 -- LET'S GO
```

Example: Script

```
Jerry: Where are we going?
To the restaurant?
Elaine (to herself): Whatever...
Kramer’s appartment
==============================
Kramer: Um, let’s go now! <i>angrily</i>
```

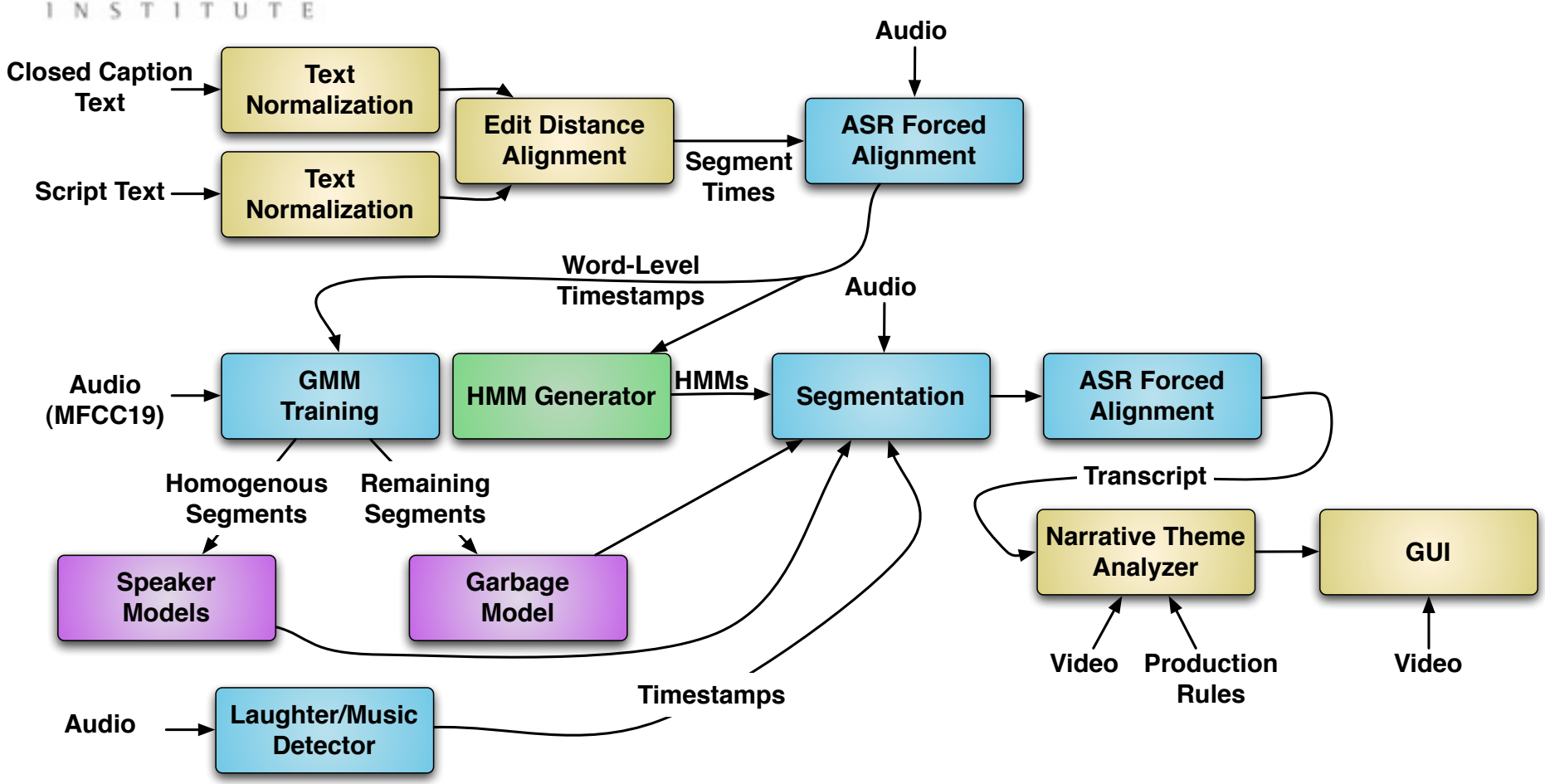
Problems

- Close Caption:
  - No speaker attribution
  - Not an exact transcription of speech
  - OCR errors
- (Fan-generated) Scripts:
  - No timings
  - Usually no strict format
  - Dialog mixed with meta-info
- Both:
  - No indication for laughter (punchlines)

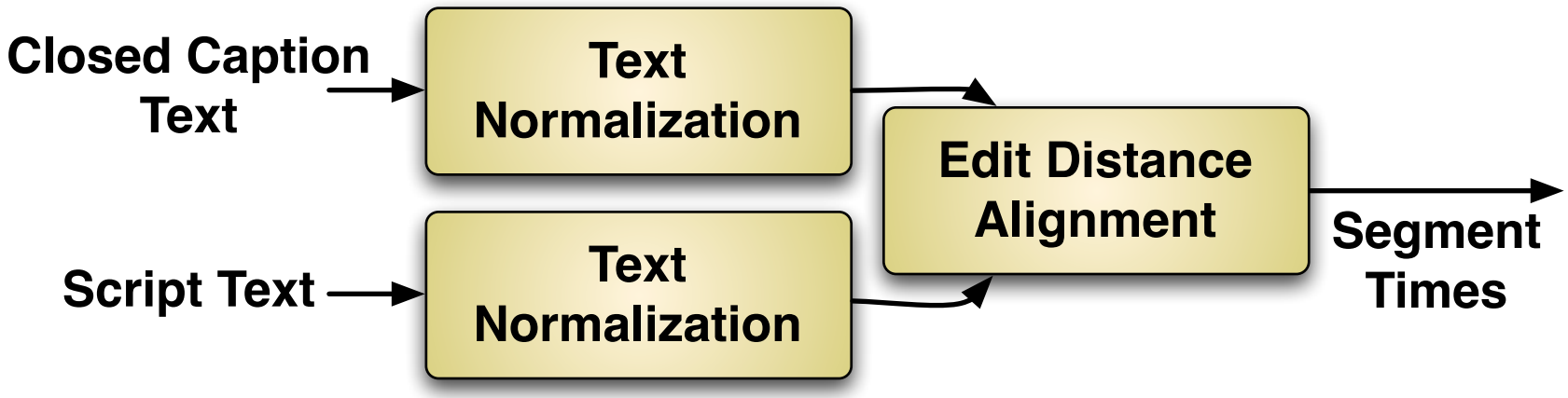
Idea

Combine closed captioning and scripts by aligning them with the audio stream and correct errors by inference.

Joke-o-Mat 2.0 Overview



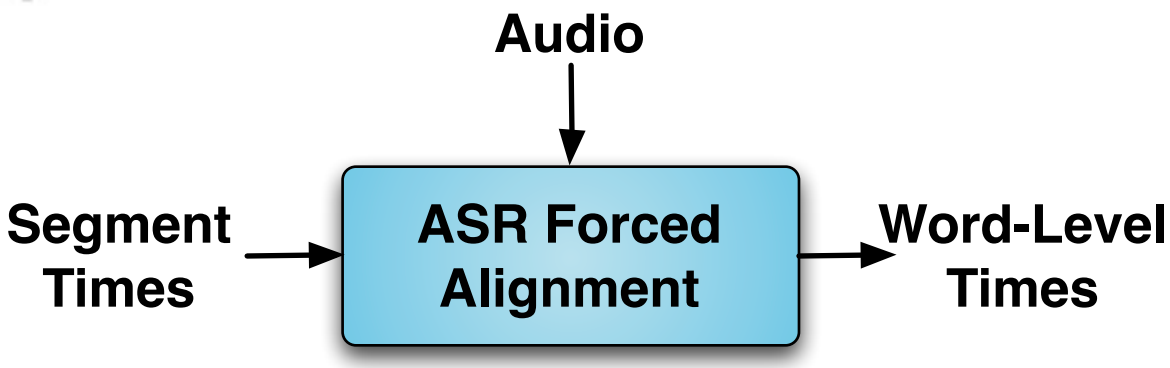
Text Normalization and Alignment



Output: Segment start/end for script text iff begining and endwords align.

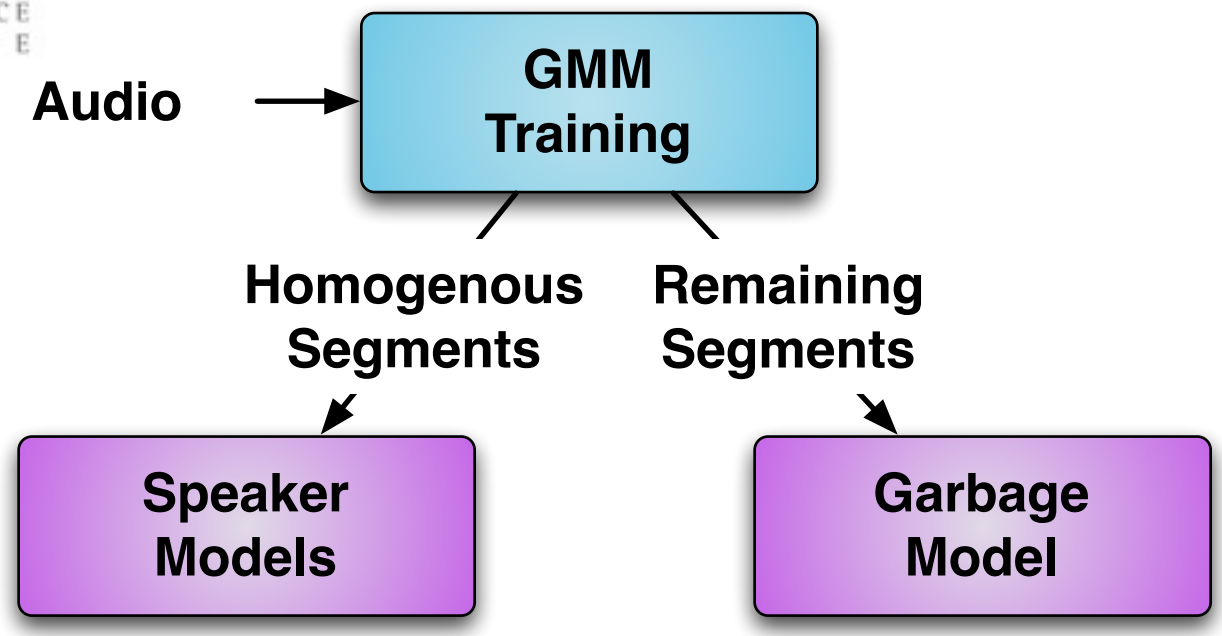
Problem: Intrasegmental speaker changes.

ASR Forced Alignment I



Only "clear" segments considered

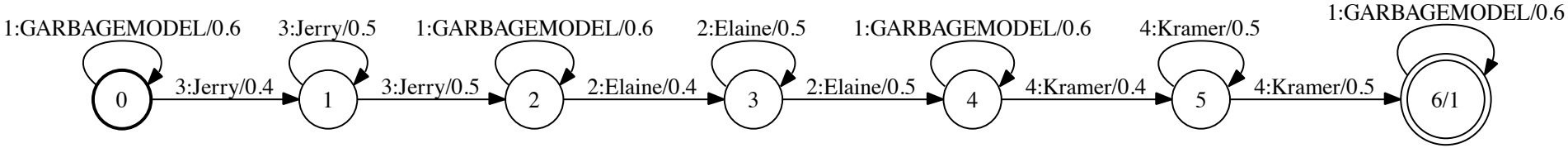
GMM Training



Input: Word-level timestamps of "clear" segments, MFCC19 features (speaker ID)

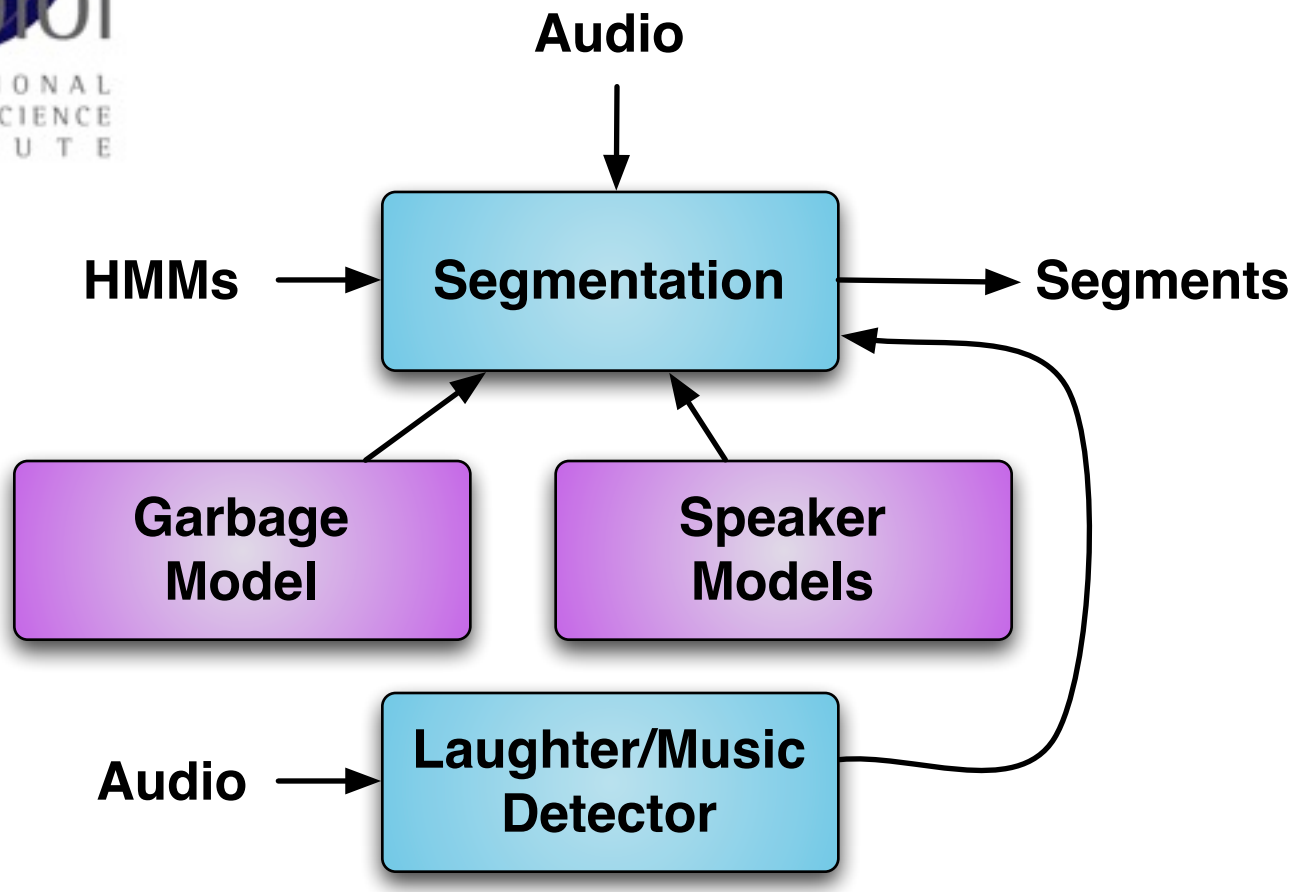
HMM Generation

Input: Word-level timestamps of "unclear" segments



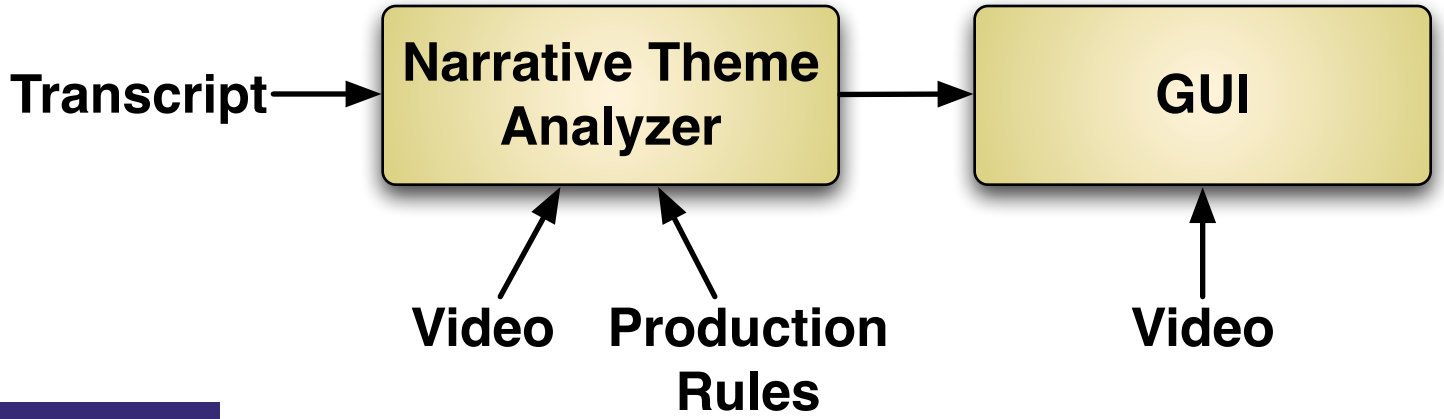

```
10.01 -> 15.22
Jerry: Where are we going?
To the restaurant?
Elaine: Whatever...
Kramer: Um, let's go now!
```

Segmentation



Final Steps

Second ASR forced alignment with better models



Result

- Only limited training data required (laughter, music)
- Automatic labeling of actor names
- Almost perfect key-word level search
- In general: Case study for how to use found data

Evaluation

Prefer Fan-augmented	16 %
Prefer Expert-generated	12 %
No Preference	72 %

Turing test-like web interface study with 25 random users

Demo

Problems Remaining

- Sometimes not enough training data for supporting actors
- Genre/Format-specific ontologies: What if there's no laugh track?
- What if there is no script/close captioning?

Future Work

Other ideas

- Incorporate user feedback (social navigation)
- Find cross-links between episodes
- “Unweave” narrative threads

Acknowkledgements

Major work on Joke-o-Mat (especially 2.0):

- Adam Janin, Luke Gottlieb

“Turing Test” evaluation:

- People from Appscio.com and Credit.com

as well as various jurors and reviewers for constructive feedback!

INTERNATIONAL COMPUTER SCIENCE INSTITUTE

Thank You! Questions?

www.icsi.berkeley.edu/jokeomat