CS 224S / Linguist 285
Spoken Language Processing

Andrew Maas | Stanford University | Spring 2025

Lecture 12: Social Meaning Extraction with ML & Course Projects

Logistics

- **Monday's lecture is a remote speaker on Zoom.**
  - You may watch only the Zoom/Panopto (ideally live to ask questions in Zoom chat)
  - Lecture hall will have the zoom + Andrew, so attend in person as usual if you prefer
  - Canvas quizzes will continue to be mentioned in lecture. Completing the quiz in 24 hours is to check for participation
- **Course project time!**
  - Homework 4 is the "default project". You will fine tune whisper and attain competitive ASR results on multiple languages with space to experiment with model and data variations
  - Do homework 4 unless you are interested in a specific topic or already working towards related research
  - To do a project:
    - Submit a proposal by Monday 5/12. Short (500 words). Just to make clear you have a feasible initial idea
    - Staff will review/revert proposals fast. Approved projects can ignore homework 4 and do a project instead
    - Project deliverables: In-class (or recorded video) spotlight talk. Final paper. Details on course website

Outline

- Course project overview

- Social meaning extraction as supervised ML
  - Bias risks and ethics when developing ML models in speech
  - Emotion recognition task definition
- Using ML responsibly in speech systems

- Appendix: Case studies in ML to model speech. Flirting. Intoxication.

Course Project Overview

Course Project Goals

- A substantial piece of work related to topics specific to this course
- A successful project result:
  - Experiments for a conference paper submission if academically oriented -or-
  - Functioning system demo. A portfolio item for job interviews, or a system you put into production!
- Reflects deeper understanding of SLProc technology than simply applying existing API's for ASR, voice commands, etc.
- **The real goal: Build something you're proud of**

A Successful Project

- Course-relevant topic. Proposed experiments or system address a challenging, unsolved SLP problem
- Proposes and executes a sensible approach informed by previous related work
- Performs error analysis to understand what aspects of the system are good/bad
- Adapts system or introduces new hypotheses/components based on initial error analysis
- Goes beyond simply combining existing components / tools to solve a standard problem

Complexity and Focus

- SLP systems are some of the most complex in AI
- Example: a simple voice command system contains:
    - Speech recognizer
      (Language model, pronunciation lexicon, acoustic model, decoder, lots of training options)
    - Intent/command slot filling (some combination of lexicon, rules, and ML to handle variation)
- Get a complete baseline system first. Then focus on improving/evaluating a specific piece
    - Do not over-invest in individual components without having a complete first version of a system or experiment pipeline
    - Focus on a subset of all areas to make a bigger contribution there. APIs/tools are a great choice for areas not directly relevant to your focus

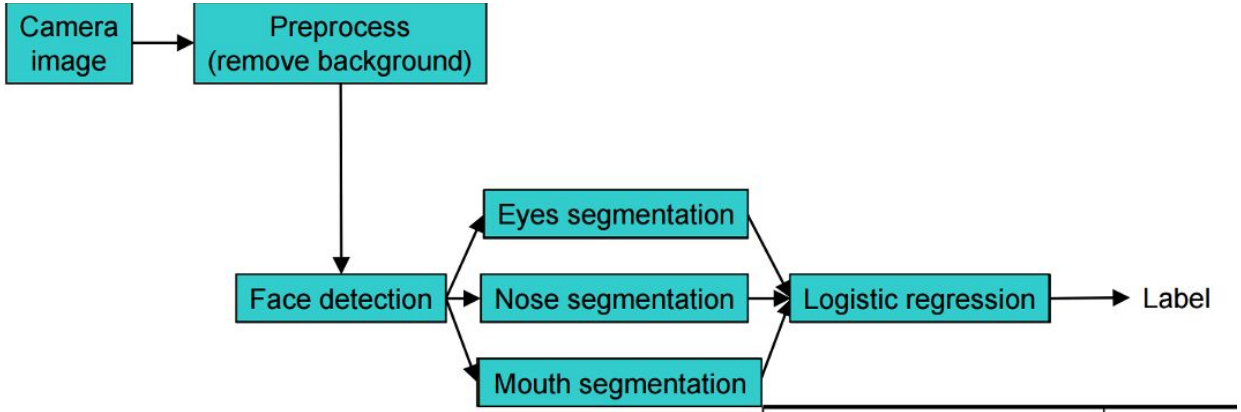
Balancing Scale and Depth

- Working on “real” scale datasets/problems is a plus…
- But don’t let scale distract from getting to the meat of your technical contribution
- Example:
    - Comparing some neural architectures for end-to-end speech recognition
        - Case 1: Use WSJ. Medium sized corpus, read speech. SOTA error rates ~3%
        - Case 2: Use Switchboard: Large, conversational corpus. SOTA error rates ~15%
- Case 2 stronger overall if you run the same experiments / error analysis. Don’t let scale prevent “thoughtful loops

Thoughtful Loops

- A single loop:
  - Try something reasonable with a hypothesis
  - Perform error analysis to investigate your hypothesis
  - Propose a modification / new experiment based on what you find
  - Try it!
  - Repeat above
- A successful project does this at least once
- Scale introduces risk of overly slow loops
- Ablative analysis or oracle experiments are a great way to guide what system component to work on

Oracle Experiments



How much error is attributable to each of the components?

Plug in ground-truth for each component, and see how accuracy changes.

Conclusion: Most room for improvement in face detection and eyes segmentation.

Component	Accuracy
Overall system	85%
Preprocess (remove background)	85.1%
Face detection	91%
Eyes segmentation	95%
Nose segmentation	96%
Mouth segmentation	97%
Logistic regression	100%

Figure: Andrew Ng's CS229 lecture on applying ML [Lecture]

Ablation Experiments

- Error analysis tries to explain the difference between current performance and perfect performance
- Ablative analysis tries to explain the difference between some baseline (much poorer) performance and current performance
- E.g., Suppose that you've build a good anti-spam classifier by adding lots of clever features to logistic regression:
    - Spelling correction
    - Sender host features
    - Email header features
    - Email text parser features
    - Javascript parser
    - Features from embedded images
- Question: How much did each of these components really help?

From: Andrew Ng's CS229 lecture on applying ML Lecture

Ablation Experiments

- Simple logistic regression without any clever features get 94% performance.
- Just what accounts for your improvement from 94 to 99.9%?
- Ablative analysis: Remove components from your system one at a time, to see how it breaks.

Component	Accuracy
Overall system	99.9%
Spelling correction	99.0
Sender host features	98.9%
Email header features	98.9%
Email text parser features	95%
Javascript parser	94.5%
Features from images	94.0%

[baseline]

- Conclusion: The email text parser features account for most of the improvement.

From: Andrew Ng's CS229 lecture on applying ML Lecture

Pitfalls in Project Planning

- Data!
  - What dataset will you use for your task?
  - If you need to collect data, why?
  - Understand that a project with a lot of required data collection creates high risk of not being able to execute enough loops
  - **Do you really need to collect data? Really?**
- Overly complex baseline system
- Relying on external tools to the point that connecting them becomes the entire effort and makes innovation hard
- Off-topic. Could this be a CS 229 project instead?

Deliverables

- All projects
  - **Proposal:** What task, dataset, evaluation metrics and approach outline?
  - **Spotlight Talk:** ~2 minute talk on your project motivation, main results, and experiments/system
  - **Final paper:** Methods, results, related work, conclusions. Should read like a conference paper
- Audio/Visual material
  - Include links to audio samples for TTS. Screen capture videos for dialog interactions (spoken dialog especially)
  - Much easier to understand your contribution this way than leave us to guess. Even if it doesn't quite work

Leveraging Existing Tools

- Free to use any tool, but realize using the Google speech API does not constitute ‘building a recognizer’
- Ensure the tool does not prevent trying the algorithmic modifications of interest (e.g. can’t do acoustic model research on speech API’s)
- Projects that combine existing tools in a straightforward way should be avoided
- Conversely, almost every project can and should use some form of tool:
    - PyTorch, Tensorflow, speech API, language model toolkit, Alexa Skills Kit, SpeechBrain, Whisper, Pre-trained embedding models, etc.
- Use tools to focus on your project hypotheses

Error Analysis with Tools

- **Project write up / presentation should be able to explain:**
  - What goal does this tool achieve for our system?
  - Is the tool a source of errors? (e.g. oracle error rate for a speech API)
  - How could this tool be modified / replaced to improve the system? (maybe it is perfect and that's okay)
- **As with any component, important to isolate sources of errors**
- **Work with tools in a way that reflects your deeper understanding of what they do internally (e.g. n-best lists)**

Sample project areas and research trends

Research Trend: Convergence of Text NLP and Spoken NLP

- **Can you simplify spoken language modeling to close the text/spoken gap?**
- **Recipe for NLP tasks:**
    - Transformer models
    - Self-supervised pre-training on large data
    - Fine tune / adapt to particular tasks

Dialog Systems

- Build a dialog system for a task that interests you (bartender, medical guidance, chess)
- Must be multi-turn. Not just voice commands or single slot intent recognizers
- Evaluation is difficult, likely will have to collect any training data yourself
- Don’t over-invest in knowledge engineering
- Lots of room to be creative and design interactions to hide system limitations
- More difficult to publish smaller scale systems, but make for great demos / portfolio items

Speech Recognition

- Benchmark corpus (WSJ, Switchboard, noisy ASR on CHIME, Librispeech)
- Establish a baseline system (ok to use pre-trained)
- Template very amenable to publication in speech or machine learning conferences
- Can be very difficult to improve on state of the art. Systems can be cumbersome to train
- Lots of algorithmic variations to try
- Successful projects do not need to improve on best existing results
- Adapting pre-trained neural approaches to new domains/tasks is great!

Speech Synthesis

- Blizzard challenge provides training data and systems for comparison
- Evaluation is difficult. No single metric
- Matching state of the art can be very tedious signal processing
- Open realm of experiments to try, especially working to be expressive or improve prosody
- Relatively large systems for SOTA. Recent deep learning approaches for easier testing

Deep Learning Approaches

- Active area of research for every area of SLP
- Beware:
  - Do you have enough training data compared to the most similar paper to your approach?
  - Do you have enough compute power / GPUs?
  - How long will a single model take to train? Think about your time to complete one ‘loop’
- Ensure you are doing SLP experiments not just tuning neural nets for a dataset

Extracting Information From Speech

- Beyond transcription, understanding emotion, accent, or mental state (intoxication, depression, Parkinson's etc.)
- Very dataset dependent. How will you access labeled data to train a system?
- Can't be just a classifier. Need to use insights from this course or combine with speech recognition
- Exploring foundation model features is acceptable
- Should be spoken rather than just written text
- Often most exciting when the dataset/task is meaningful

Project summary

- Have fun. Build something you're proud of
- Post on Ed or use Andrew + Tolúlọpẹ́ office hours for early project feedback
- Proposals due soon! (Proposal should be easy to write if you have a project formulation in mind)

- Post on Ed to find shared interests for project groups

Extracting Social Meaning with Supervised ML

Research Trend: Foundation Model Features

- Large, pre-trained deep learning models provide useful features for many spoken tasks
- Lots to explore in this area in ASR, TTS, dialog, and multi-modal
- Pre-trained models great for rapid progress on projects and smaller training sets
- Popular models with available features:
    - Wav2Vec 2.0
    - HuBERT
    - HuggingFace has several more!

Research Trend: Foundation Model Features

Model	Speech	Input format	Framework	Encoder	Loss	Inspired by
LIM [36]	✓	raw waveform	(d)	SincNet	BCE, MINE or NCE loss	SimCLR
COLA [36]	✗	log mel-filterbanks	(d)	EfficientNet	InfoNCE loss	SimCLR
CLAR [33] (semi)	✗	raw waveform log mel-spectrogram	(d)	1D ResNet-18 ResNet-18	NT-Xent + cross-entropy	SimCLR
Fonseca et al. [36]	✗	log mel-spectrogram	(d)	ResNet, VGG, CRNN	NT-Xent loss	SimCLR
Wang et al. [88]	✗	raw waveform + log mel-filterbanks	(d)	CNN ResNet	NT-Xent loss + cross-entropy	SimCLR
BYOL-A [89]	✗	log mel-filterbanks	(b)	CNN	MSE loss	BYOL
Speech2Vec [48]	✓	mel-spectrogram	(a)	RNN	MSE loss	Word2Vec
Audio2Vec [91]	✓✗	MFCCs	(a)	CNN	MSE loss	Word2Vec
Carr [67]	✓	MFCCs	(a)	Context-free network	Fenchel-Young loss	-
Ryan [68]	✗	constant-Q transform spectrogram	(a)	AlexNet	Triplet loss	- -
Mockingjay [92]	✓	mel-spectrogram	(a)	Transformer	L1 loss	BERT
TERA [93]	✓	log mel-spectrogram	(a)	Transformer	L1 loss	BERT
Audio ALBERT [94]	✓	log mel-spectrogram	(a)	Transformer	L1 loss	BERT
DAPC [95]	✓	spectrogram	(a)	Transformer	Modified MSE loss + orthogonality penalty	BERT
PASE [96]	✓	raw waveform	(a)	SincNet + CNN	L1, BCE loss	BERT
PASE+ [97]	✓	raw waveform	(a)	SincNet + CNN + QRNN	MSE, BCE loss	BERT
CPC [40]	✓	raw waveform	(a)	ResNet + GRU	InfoNCE loss	-
CPC v2 [59]	✓	raw waveform	(a)	ResNet + Masked CNN	InfoNCE loss	-
CPC2 [98]	✓	raw waveform	(a)	ResNet + LSTM	InfoNCE loss	-
Wav2Vec [84]	✓	raw waveform	(a)	1D CNN	Contrastive loss	-
VQ-Wav2Vec [85]	✓	raw waveform	(a)	1D CNN + BERT	Contrastive loss	BERT
Wav2Vec 2.0 [81]	✓	raw waveform	(a)	1D CNN + Transformer	Contrastive loss	BERT
HuBERT [99]	✓	raw waveform	(c)	1D CNN + Transformer	Contrastive loss	BERT

Table: An overview of the recent audio self-supervised learning methods. The "speech" column distinguishes whether a method addresses speech tasks or for general purpose audio representations. (Liu *et al*, 2022)

Supervised ML as an Analysis Tool

- Define your task / hypothesis (the dataset and ML task define the problem)
  - e.g. detecting alcohol intoxication in speech
- Collect/access data and annotations (y) for your task.
- Ensure data is representative for your problem
- Define and test your modeling approach and inputs (x)
  - Optimize to find x,f() to improve f(x)=y for your data
- Analyze results, feature importance, model weights etc.
  Use the model to understand relationships in your data
  Requires careful attention to modeling assumptions and causality/correlation

Supervised ML as an Analysis Tool

- **Define a dataset to predict outcome variable from speech+text inputs. Build and analyze ML models to investigate predictive associations of input/outcome variables**

- **Framework is highly flexible.**
  Requires careful decisions about datasets and what conclusions to draw

- **Use cases:**
  - Accurately recognize y in a live/ongoing system
    - Validate x is reasonable/ethical. Build best possible f(x). Favor accuracy
  - Infer findings/relationships predictive of y
    - Careful analysis of x, f(x). Favor interpretability

ML Development projects generally

What are the steps for a complete ML application?

1. Define the function to approximate (inputs and outputs). f(x) -> y
2. Integration with broader products and services
3. Data sources for system building and ongoing usage
4. Evaluation with business and development metrics
5. Requirements for ongoing service or repeated analysis
6. Success criteria

Preventing negative impacts in ML starts with project design

Bias and misuse often is using ML wrong

- **Many issues stem from choosing where to use ML and ensuring the ML task has plausible inputs/outputs**

- **Where can bias come from? In ML lifecycle steps:**
  - “Data sources for system building and ongoing usage”
  - “Evaluation with business and development metrics”
  - Defining labels and metrics implicitly defines blindspots

Emotion recognition task design. Defining output labels

Given input utterance (audio, usually with text transcript) predict different aspects of emotion:

Arousal: The intensity or energy level of an emotion, ranging from calm to excited

Valence: The positive or negative nature of the emotion, from pleasant to unpleasant.

Dominance: The sense of control or power one feels in a situation, varying from feeling submissive to dominant.

(Example dataset: IEMOCAP. 12 hours of acted emotional speech videos)

Separate task: Sentiment analysis tasks. (E.g. MOSI dataset: Subjectivity, sentiment intensity, per-frame, and per-opinion annotated visual/audio frames. Data from YouTube with a focus on vlogs for real-world distribution of expressive monologue speakers

ParaLBench task suite

Cover several task definitions for emotion/sentiment

Also varying data collection conditions (conversational, acted, monologues, etc.)

Ideally audio analysis pipelines work for many task and data conditions

TABLE II
A TAXONOMY OF THE SELECTED PARALINGUISTIC TASKS FOR PARALBENCH.

Taxonomy	Tasks	Datasets	# Classes
Short-term	Emotion	MELD	7
	Emotion	MSP-Podcast	5
	Emotion	IEMOCAP	4
	Arousal	IEMOCAP	1*
	Valence	IEMOCAP	1*
	Dominance	IEMOCAP	1*
	Sentiment	MELD	3
	Sentiment	CMU-MOSI	2
Medium-term	Sarcasm	MUStARD	2
	Influenza	FluSense	9
	Stutter	SEP-28K	2
	Depression	DAIC-WOZ	2
Long-term	Gender	MSP-Podcast	2
	Age	VCTK	1*
	Accent	VCTK	11
	Dialect	TIMIT	8

* Regression only

(ParaLBench, 2024)

ParaLBench experiment design

Compare many encoder models for audio by creating a standard feature + classifier pipeline

Run many experiments to compare input features for fixed dataset + labels

Question: Which audio representation yields the best classification results?

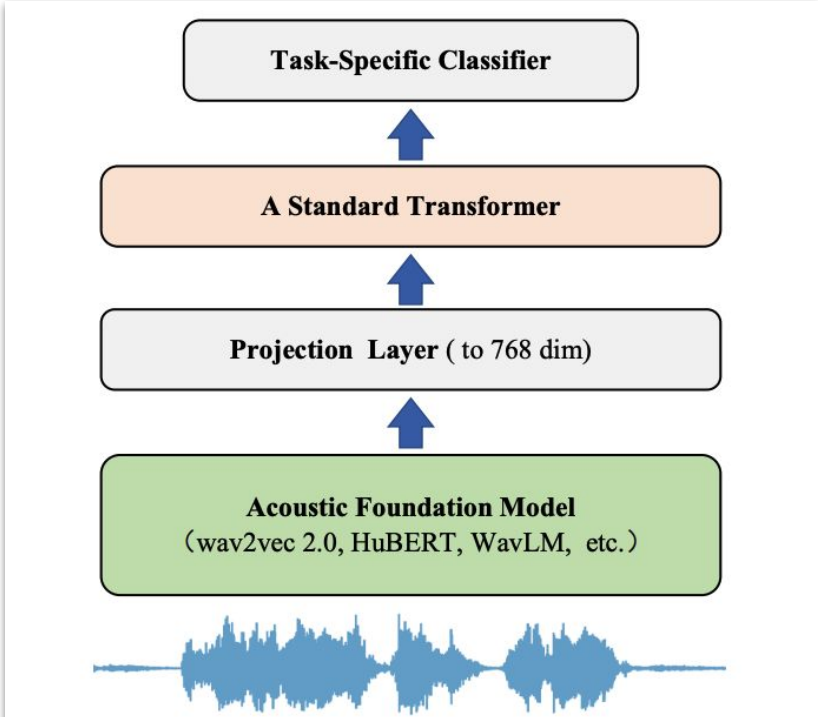


Fig. 2. An evaluation diagram for ParaLBench. It first extracts deep representations from audio signals through an acoustic foundation model. These representations are then projected to the ones with a fixed dimension and sequentially fed into a standard Transformer. Finally, a task-specific classifier (i. e., two dense layers) is appended for classification.

(ParaLBench, 2024)

ParaLBench task suite

Using standard encoder + classifier experiment design, compare many encoder and hand-built feature sets

Weighted average (WA), Unweighted average (UA) and Weighted F1 (WF1) give slightly different views

TABLE IV
RESULTS OF THE SELECTED *short-term* PARALINGUISTIC TASKS (I. E., EMOTION [EMO.] AND SENTIMENT [SENT.]) IN PARALBENCH OVER TWO HANDCRAFTED FEATURE SETS AND 14 ACOUSTIC FOUNDATION MODELS.

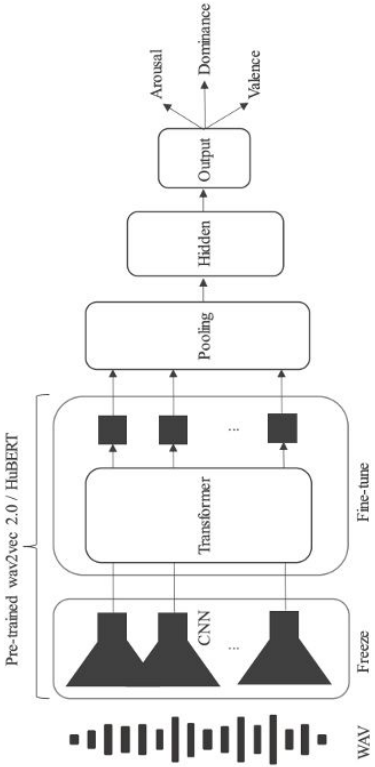
Foundation models	MELD (Emo., 7-cls)			MSP-Podacst (Emo., 5-cls)			MELD (Sent., 3-cls)			CMU-MOSI (Sent., 2-cls)		
	WA	UA	WF1	WA	UA	WF1	WA	UA	WF1	WA	UA	WF1
eGeMAPS	.481	.143	.313	.502	.218	.379	.491	.347	.351	.531	.527	.535
ComParE-2016	.481	.143	.312	.475	.200	.307	.481	.333	.312	.464	.537	.387
CLAP-HTSAT	.482	.146	.319	.514	.243	.455	.484	.356	.386	.528	.534	.532
data2vec-audio-base	.498	.176	.367	.559	.294	.507	.499	.380	.420	.641	.615	.635
emotion2vec-base	.531	.247	.465	.586	.342	.546	.561	.503	.547	.711	.676	.699
HuBERT-base	.481	.216	.420	.579	.347	.538	.519	.434	.483	.676	.659	.675
HuBERT-large	.514	.209	.419	.604	.364	.567	.535	.428	.471	.585	.581	.588
wav2vec-large	.464	.149	.333	.475	.200	.306	.470	.358	.397	.583	.557	.578
wav2vec2-base	.481	.143	.312	.527	.253	.449	.481	.333	.312	.585	.581	.588
wav2vec2-large	.481	.143	.312	.510	.254	.415	.482	.335	.315	.580	.583	.584
wav2vec2-large-age-gender	.488	.235	.427	.580	.315	.531	.534	.412	.459	.574	.591	.576
wav2vec2-large-xlsr-53	.482	.144	.314	.573	.250	.471	.489	.344	.339	.583	.557	.576
WavLM-base	.488	.208	.416	.588	.332	.546	.528	.443	.496	.646	.613	.635
WavLM-large	.515	.262	.470	.615	.363	.576	.575	.480	.539	.710	.711	.712
Whisper-base	.514	.249	.462	.608	.348	.567	.553	.454	.510	.662	.662	.665
Whisper-large	.519	.238	.443	.606	.375	.574	.561	.458	.513	.672	.680	.675

(ParaLBench, 2024)

Emotion Recognition with Foundation Model Features

Table: State-of-the-art 4-class emotion recognition performance on IEMOCAP using transformer-based architectures ranked by unweighted average recall (UAR) / weighted average recall (WAR). The table encodes whether the base or large (L) architecture was used as well as whether the pre-trained model was fine-tuned for speech recognition (FT-SR). The column FT-D marks if the transformer layers were further fine-tuned during the downstream classification task.

(Wagner *et al*, 2022)



	Work	Model	L	FT-SR	FT-D	UAR	WAR
1	Krishna [27]	w2v2-L	✓			60.0	
2	Yuan *et al.* [28]*	w2v2-L	✓			62.5	62.6
3	Wang *et al.* [23]	w2v2-b					63.4
4	Yang *et al.* [29]	w2v2-b				63.4	
5	Pepino *et al.* [30]	w2v2-b		✓		63.8	
6	Wang *et al.* [23]	hubert-b					64.9
7	Yang *et al.* [29]	hubert-b				64.9	
8	Wang *et al.* [23]	w2v2-L	✓				65.6
9	Yang *et al.* [29]	w2v2-L	✓			65.6	
10	Pepino *et al.* [30]	w2v2-b				67.2	
11	Wang *et al.* [23]	hubert-L	✓				67.6
12	Yang *et al.* [29]	hubert-L	✓			67.6	
13	Chen and Rudnicky [31]	w2v2-b			✓	69.9	
14	Makiuchi *et al.* [32]	w2v2-L	✓			70.7	
15	Wang *et al.* [23]	w2v2-b		✓	✓		73.8
16	Chen and Rudnicky [31]	w2v2-b			✓	74.3	
17	Wang *et al.* [23]	hubert-b			✓		76.6
18	Wang *et al.* [23]	w2v2-L	✓	✓	✓		76.8
19	Wang *et al.* [23]	w2v2-b			✓		77.0
20	Wang *et al.* [23]	w2v2-L	✓		✓		77.5
21	Wang *et al.* [23]	hubert-L	✓	✓	✓		79.0
22	Wang *et al.* [23]	hubert-L	✓		✓		79.6

* For a fair comparison we report the result on utterance-level. Authors report better performance on phonetic level, though.

Defining “Foundation Model”

Given an ML task with input feature vector x and output y

We use a pre-existing model Φ to encode x and obtain $\Phi(x) \rightarrow z$

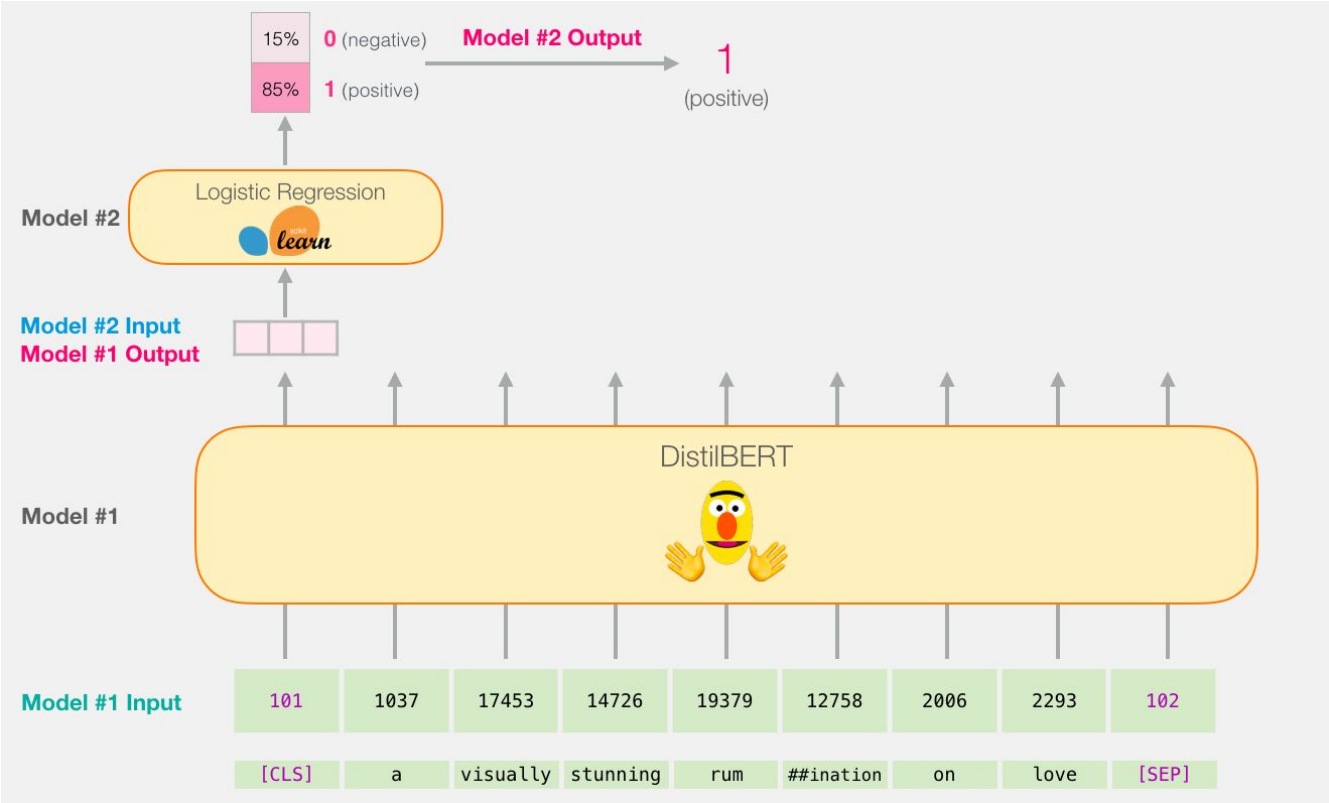
We transform the entire dataset X and build a supervised learning model for examples (z, y) to obtain a supervised predictor F where

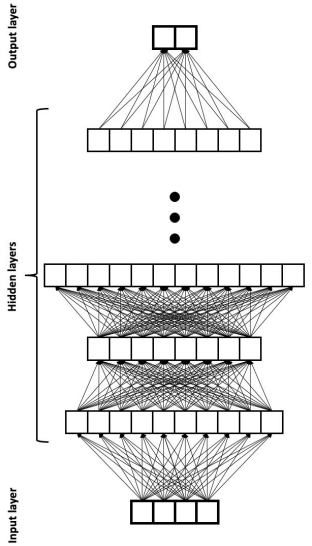
$F(z) = F(\Phi(x)) \rightarrow y$

Why “foundation”? Refers to feature encoder models that are typically derived from large, expensive deep learning models

Often we expect foundation features to help with many specific tasks that operate on the same input feature domain (e.g. text, images, audio)

Using foundation features: BERT encoder for text data





Visual BERT guide

Impact of fine tuning using LoRA

Fine tuning the base model using a low-rank adapter (LoRA) improves performance.

Many audio models can benefit from task-specific training since audio tasks can vary based on labels, data collection settings, and language.

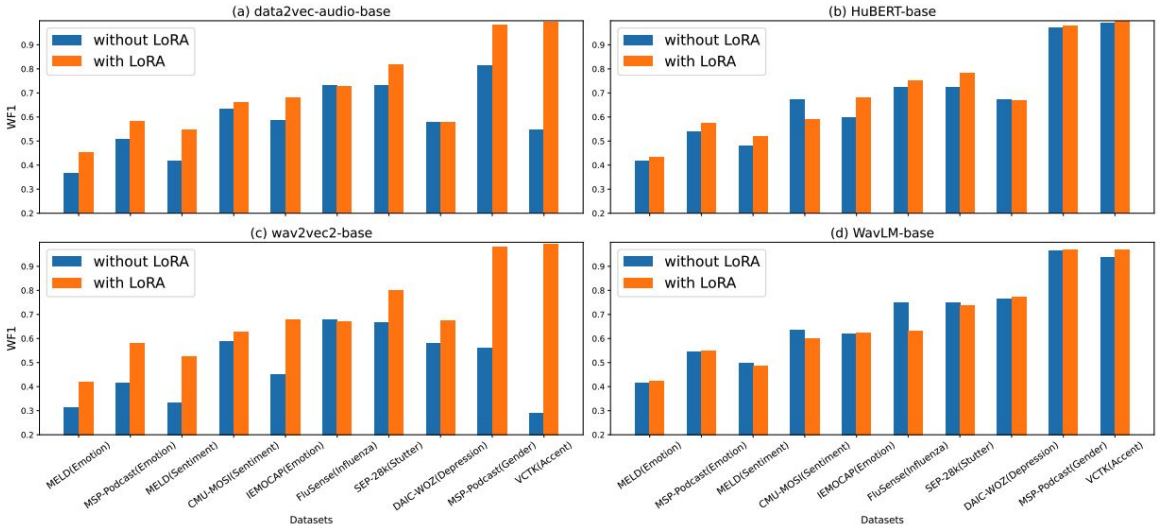


Fig. 4. Efficient fine-tuning of the acoustic foundation models using LoRA.

(ParaLBench, 2024)

Questions in Emotion Recognition

- **How do we know what emotional speech is?**
  - Acted speech vs. natural (hand labeled) corpora
- **What can we classify? Usually best to customize this for your specific needs/product**
  - Distinguish among multiple ‘classic’ emotions
  - Distinguish emotive aspects:
    - Valence: is it positive or negative?
    - Activation: how strongly is it felt? (a bit sad vs despair)
    - Dominance: Is speaker dominant vs submissive in conversational stance?
    - Others: Sentiment categories + arousal/intensity level
- **What features best predict emotions?**
  Probably pre-trained audio encoders of some form
- **What techniques best to use in classification?**
  Simple linear models or MLPs work with good features

Slide: Julia Hirschberg

Social Signal Processing = Affect/emotion labels or tasks

- Detecting frustration of callers to a help line
- Detecting stress in drivers or pilots
- Detecting depression, intoxication
- Detecting interest, certainty, confusion in online tutors
  - Pacing/Positive feedback
- Hot spots in meeting summarizers/browsers

Detecting or monitoring health biomarkers from speech

Datasets and task definitions require precision when building/deploying healthcare-related systems

It's possible to detect various cognitive disorders and depression+anxiety from speech

Deploying such systems requires careful integration with care providers or downstream systems

Example: Classifying cognitive impairment given audio from prompted speech (PROCESS challenge, 2024)

TABLE I
BASELINE CLASSIFICATION RESULTS FOR THE COOKIE THEFT (CT), SEMANTIC FLUENCY AND PHONEMIC FLUENCY (VF) AND THEIR COMBINATION (CT+VF) ARE PRESENTED IN TERMS OF ACCURACY (ACC.), PRECISION (PREC.), RECALL (REC.), AND F1-SCORE, REPORTED AS PERCENTAGES.

Model	Prompt	Acc.	Prec.	Rec.	F1
SVC (eGeMAPS)	CT	57.5	53.5	**61.2**	**55.0**
	VF	45.0	38.8	39.1	38.3
	CT+VF	50.0	41.7	42.3	41.7
RFC (eGeMAPS)	CT	**60.0**	**71.7**	49.9	53.3
	VF	52.5	33.1	35.9	33.9
	CT+VF	52.5	66.1	43.9	47.4
RoBERTa-Classifier	CT	52.5	36.1	39.7	36.8
	VF	55.0	35.6	38.1	35.6
	CT+VF	52.5	32.2	35.9	32.9

Using ML responsibly in speech systems

What causes negative impacts of ML?

- Harmful system: negative impact on people or the world
- Biased system: performs differently, or not at all, for some subpopulation of inputs.
- *We are not worried about sentient, runaway ML models*

Common causes ML system negative impacts

- System design issues
- Insufficiently trained / low performing ML models
- Distribution shift and mis-application

System design issues

- Problem formulation / ML task should be ethical
- Depends little on particular ML model

- Example:
  - *Predict future criminality from face image*

  - *TikTok recommendations. Using predictions of "will viewer click this?" with no content filtering can harm kids*

Mitigating system design issues

- Engage *all* stakeholders during development
- **Transparency** of system design and data provenance
- **Accountability**: Who in an organization is responsible for ML performance?
- Further reading: New York City Gov't AI Primer

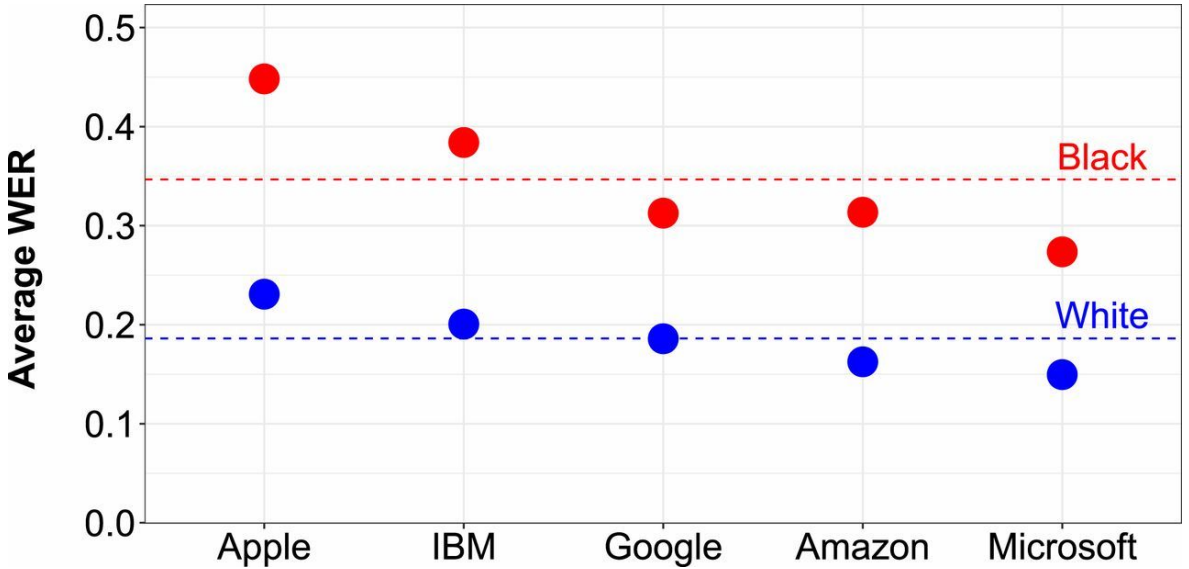
Insufficiently trained systems

- **ML is never perfect. Potential negative impacts of mistakes**
  - Or lower quality experience for some inputs/users
- **Often times ML mistakes are not uniform random**
  - Performance issues might disproportionately affect certain *subgroups* of input examples

- ***Interpretable* ML models can help guide work/debugging**
- **Targeted evaluations and model cards help define boundaries of model capabilities**

Subgroup bias

- Different/worse ML performance on subsets of inputs
- Example: Speech recognition word error rates for black vs white native US English speakers (Koeneck et al, 2020)



Subgroup bias mitigation

- Active research on detecting and patching subgroup bias
- Fundamentally comes back to (1) training data distribution and (2) generalization performance of models

- *Auditing/testing systems to check for bias is the most reliable strategy. Make it part of deployment checklist!*

Distribution shift and misapplication of models

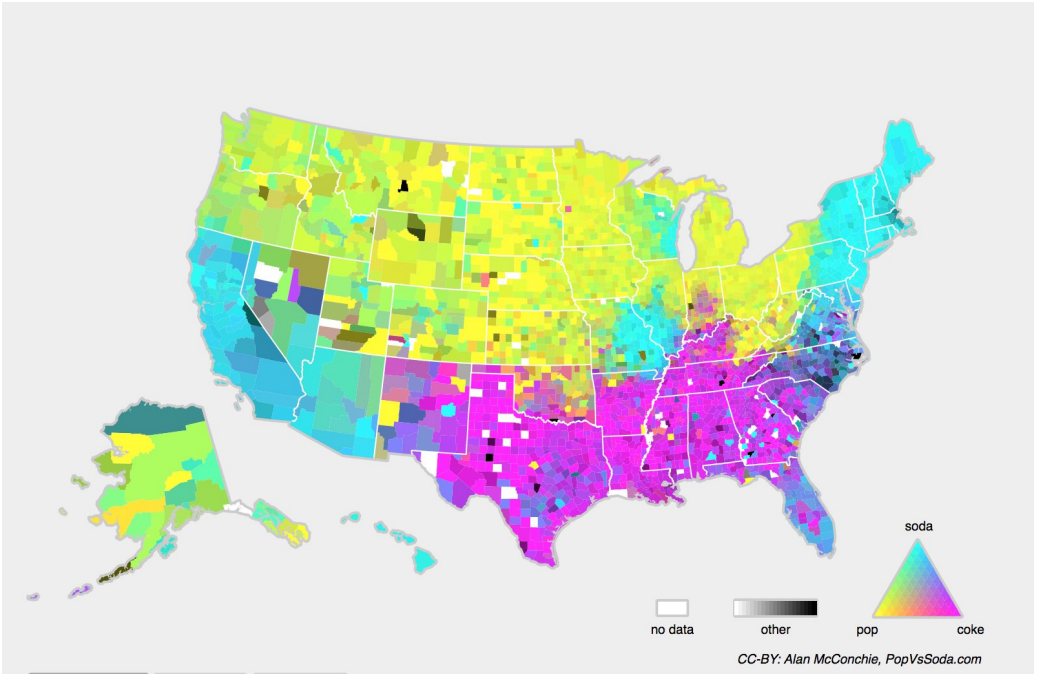
- We assume test/production data and training data are IID
  - == training set is representative of all future inputs

- Distribution shift: When test/production distribution is different or drifts from IID relative to training data

Distribution shift examples

Train

Test



Examples from [d2l.ai reading]

Distribution shift / misapplication examples

- **IBM Watson cancer diagnosis assistant ML system**
  - Trained from experts at one hospital
  - Applied to hospitals with different standards of care

- **Online platform user biometric verification via keystrokes**
  - Trained using pass phrase “My typing is my password”
  - In production, fed any pass phrase. Performance worse.

Model cards

- To aid transparency and set expectations of a model
- Includes:
  - Dataset and model used
  - Evaluations
  - Known issues / subgroup biases

- Add this as part of delivering/deploying an ML system

Model cards

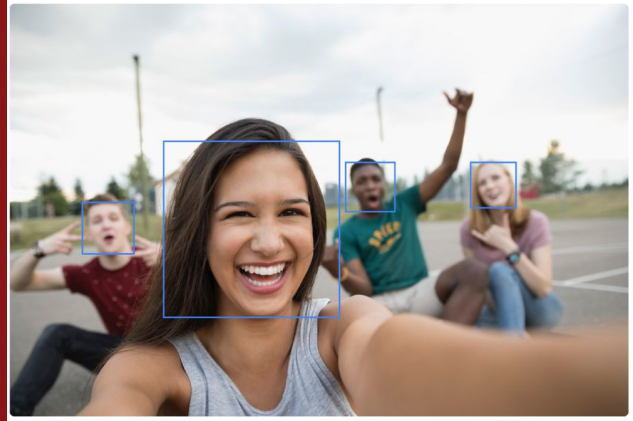
Google face detection

model card

Speech APIs don't always have clear "model card"

Evaluate on your actual data if possible to compare

MODEL DESCRIPTION



Input: Photo(s) or video(s)

Output: For each face detected in a photo or video, the model outputs:

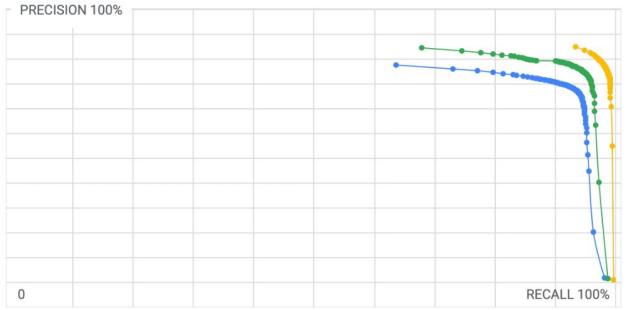
- Bounding box coordinates
- Facial landmarks (up to 34 per face)
- Facial orientation (roll, pan, and tilt angles)
- Detection and landmarking confidence scores.

No identity or demographic information is detected.

Model architecture: MobileNet CNN fine-tuned for face detection with a single shot multibox detector.

View public API documentation

PERFORMANCE



Overall model performance, and performance sliced by different image and face characteristics, were assessed, including:

- Derived characteristics (face size, facial orientation, and occlusion)
- Face demographics (human-perceived gender presentation, age, and skin tone)

Overall performance measured with Precision-Recall (PR) values and Area Under the PR Curve (PR-AUC) - standard metrics for evaluating computer vision classifiers. Download raw performance results data here.

Disaggregated performance measured with Recall, which captures how often the model misses faces with specific characteristics. Equal recall across subgroups corresponds to the "Equality of Opportunity"fairness criterion.

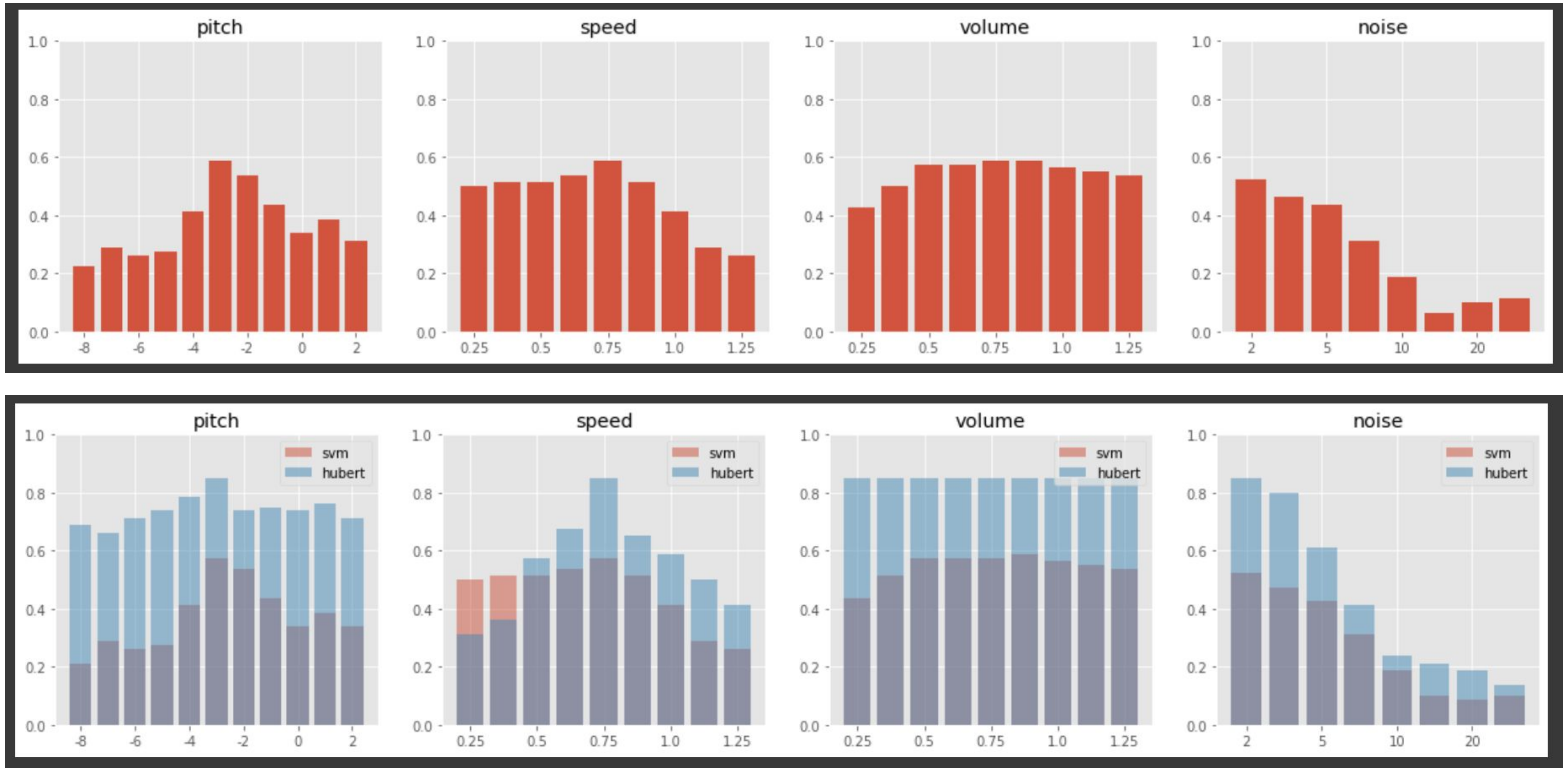
Performance evaluated on: Three research benchmarks distinct from the training set:

- A subset of Open Images
- Face Detection Data Set and Benchmark
- Labeled Faces in the Wild

Google face detection example model card

Example: Targeted evaluations. Vary speed, noise, etc. and re-measure

Audio classification tasks comparing hubert embeddings vs raw features



Summary: Building responsible ML

- Focus: Build quality ML models that do what they claim!
- Thoughtful system design
- Test for and document subgroup performance issues
- Data provenance, transparency (and privacy) is central

- **Realize that YOU may be the only informed person in critical decisions about where/how to apply ML**

Questions?

Course Project Q&A

Case study: Flirtation

Interpersonal Stance: Our Goals

- Friendliness
- Assertiveness
- Flirtation
- Awkwardness

Methodology

- Given speech and text from a conversation
- Can we tell if a speaker is
  - Awkward?
  - Flirtatious?
  - Friendly?
- Dataset:
  - 1000 4-minute “speed-dates”
  - Each subject rated their partner for these styles
  - The following segment has been lightly signal-processed

(Jurafsky, Ranganath & McFarland. 2009)

References

E.J. Finkel, P.W. Eastwick. 2008. Speed-dating. Current Directions in Psychological Science, 17 (3) (2008), p. 193

Place, S. S., Todd, P. M., Penke, L., & Asendorpf, J. B. (2009). The ability to judge the romantic interest of others. Psychological Science, 20(1), 22-26.

M.E. Ireland, R.B. Slatcher, P.W. Eastwick, L.E. Scissors, E.J. Finkel, J.W. Pennebaker. 2011. Language style matching predicts relationship initiation and stability. Psychological Science, 22 (1) (2011), p. 39

speed dating *noun*

Menu

speed dating [uncountable]

an event at which you meet and talk to a lot of different people for only a few minutes at at a time. People do this in order to try to meet someone and have a romantic relationship.

Extracting Social Meaning

- **Stance**
  - Friendly, flirt, awkward, assertive
- **Social Bond**
  - Clicking or Connection
  - Romantic Interest
- **946 4-minute dates**
  - ~800K words, hand-transcribed
  - ~60 hours, from shoulder sash recorders
  - 3 events, 20x20=400 dates x 3
  - Date perceptions, demographics, preferences

Data Annotation

- Each speaker wore a microphone
- So each date had two recordings
- The wavefile from each speaker was manually segmented into 4-minute dates
- Professional transcription service produced:
  - words, laughter, disfluencies
  - timestamps for turn beginning and end (1 second)
    - for 10% of the dates, timestamp at 0.1 second granularity
  - using both recordings

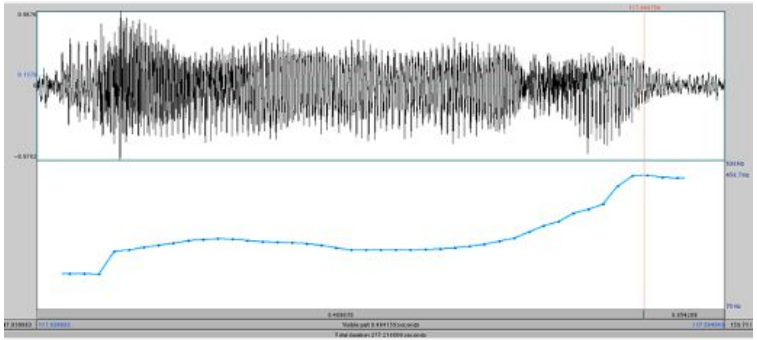
Study 1: What We Attempted to Predict

Conversational style:

- How often did they behave in the following ways on this date?
  - On a scale of 1-10 (1=never, 10=constantly)
- **awkward**
- **friendly**
- **flirtatious**
- **assertive**

Features

- **Prosodic**
  - pitch (min, mean, max, std)
  - intensity (min, max, mean, std)
  - duration of turn
  - rate of speech (words per second)
- **Lexical**
  - negation words (don't, didn't, won't, can't, not, never)
  - hedges (kind of, sort of, probably, I don't know)
  - personal pronouns (I, you, we, us)
- **Dialog**
  - Questions
  - backchannels ("uh-huh", "yeah")
  - appreciations ("Wow!", "That's great!")
  - sympathy ("That's awful!" "Oh, that sucks!")



Engineering Studies 16 Binary Classifiers

- Female ±Awkward, Male ±Awkward,
- Female ±Friendly, Male ±Friendly,
- Female ±Flirtatious, Male ±Flirtatious,
- Female ±Assertive, Male ±Assertive

- Each study run twice, on:
  - self-assessed
  - alter-assessed

- Multiple classifier experiments
  - L1-regularized logistic regression
  - SVM w/RBF kernel

Test Set

- For each of the 16 experiments
  - Sort all 946 dates
  - Choose top 10% as positive class
  - Choose bottom 10% as negative class
  - ignore 80% of dates in the middle!
- 5-fold cross-validation within this small training and test set
- Goal: distinguishing social interactants who are reported to exhibit (or not exhibit) clear social intentions or styles

Results Using SVM Classifier

Using my speech to predict what my date says about me

	Male Speaker	Female Speaker
Flirting	65%	78%
Friendly	71%	64%
Awkward	67%	67%
Assertive	65%	69%

Results Using SVM Classifier

Using my speech to predict what I say about myself

	Male Speaker	Female Speaker
Flirting	66%	74%
Friendly	76%	71%
Awkward	63%	67%
Assertive	73%	64%

What Do Flirters Do?

- **Women when flirting:**
  - raise pitch ceiling
  - talk faster
  - say “I” and “like”, use more hedges
  - laugh at themselves
- **Men when flirting:**
  - raise their pitch floor
  - laugh at their date (teasing?)
  - say “you”
  - don’t use words related to academics
  - say “um”, “I mean”, “you know”

Unlikely Words for Male Flirting

- academia
- interview
- teacher
- phd
- advisor
- lab
- research
- management
- finish

Assertive

- **Assertive men**
  - talk more
  - use more negative emotion
  - lower their pitch floor
  - use more agreements and appreciations
  - use more “um”, “you”
  - use less negation
- **Assertive women:**
  - use more negation (“no”, “didn’t”, “don’t”)
  - talk about academics
  - are less sympathetic
  - accommodate more (content words)
  - use more “I” and “I mean”
  - use less negative emotion

What Makes an Awkward Conversationalist?

- **Awkward people:**
  - use more hedges
  - ask more questions
- **Awkward men:**
  - don't talk about academics
  - do swear or use negative emotion
- **Awkward women:**
  - do talk about academics
  - talk more, and talk faster
  - don't laugh at their date
  - don't use "I"

(Actionable?) Conclusions

- How to date:
- Don’t talk about your advisor
- Focus on the empowered party
- Flirting women raise pitch ceiling – flirting men raise pitch floor

What Makes Someone Seem Friendly? “Collaborative Conversational Style”

- **Friendly people:**
  - laugh at themselves
  - don’t use negative emotions
- **Friendly men**
  - are sympathetic and agree more often
  - don’t interrupt
  - don’t use hedges
- **Friendly women:**
  - higher max pitch
  - laugh at their date

Case study: Intoxication

Holien et al., 2001: Methods

- 35 young adults, 19 males, 16 females
- Given series of doses of alcohol
- Speech collected at 4 BAC stages
  - Rainbow passage
  - Difficult words (buttercup, shapoopie)
  - Extemp speech (“Tell us about your favorite TV program)
  - Head-mounted mics
- Investigated:
  - F0 mean and variance
  - duration/rate of speech
  - Intensity
  - disfluencies

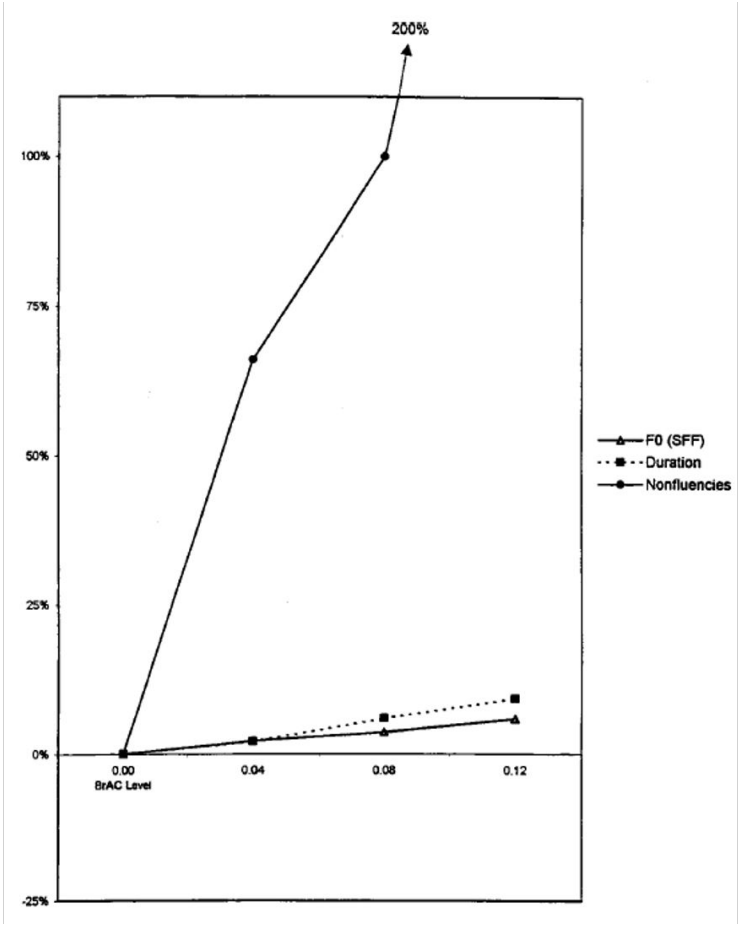
Hollien et al., 2001 Results: Duration

Group	Level of intoxication (BrAC) 0.00	0.04	0.08	0.12	Shift (0.00–0.12)
Men					
Mean (s)	25.3	25.8	26.8	27.6	+2.3
S.D. (s)	2.9	2.5	2.1	2.5	
Women					
Mean (s)	25.1	25.5	25.7	27.5	+2.4
S.D. (s)	2.2	2.2	2.4	2.7	

Hollien et al., 2001 Results: Disfluencies

Subjects	*N*	Experimental condition (BrAC) 0.00	0.04	0.08	0.12
Males	19				
Mean		3.2	4.7	6.5	8.6
SD		2.0	2.6	3.1	3.4
Females	16				
Mean		2.2	3.5	4.7	6.1
SD		1.7	2.2	2.7	3.0
Mean	35	2.7	4.1	5.6	7.4

Hollien et al., 2001 Results: Magnitudes



A Famous Case Study

Johnson, K., Pisoni, D. & Bernacki, R. (1990) Do voice recordings reveal whether a person is intoxicated?: A case study. Phonetica. 47: 215-237

Exxon Valdez oil spill

From Wikipedia, the free encyclopedia Coordinates: 60.83333°N 146.86667°W

The ***Exxon Valdez* oil spill** occurred in Prince William Sound, Alaska, on March 24, 1989, when the *Exxon Valdez*, an oil tanker bound for Long Beach, California, struck Prince William Sound's Bligh Reef and spilled 260,000 to 750,000 barrels (41,000 to 119,000 m^3) of crude oil.[1][2] It is considered to be one of the most devastating human-caused environmental disasters.[3] As

***Exxon Valdez* oil spill**

3 days after *Exxon Valdez* ran aground

Location	Prince William Sound, Alaska
Coordinates	60.83333°N 146.86667°W
Date	24 March 1989
Cause	

Was Captain Hazelwood Drunk?

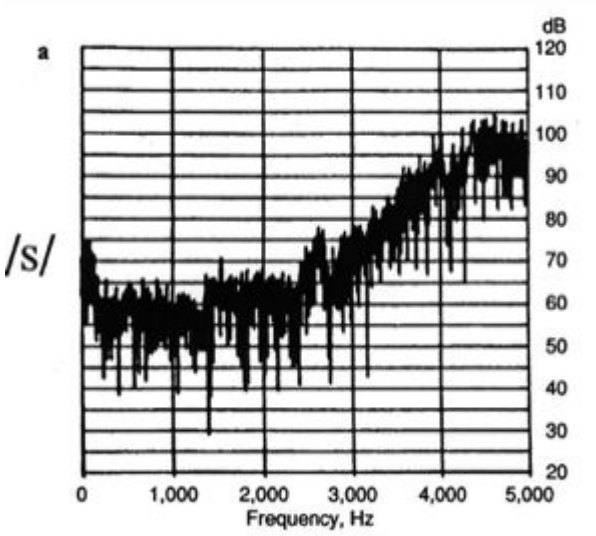
- Not clear if this is relevant, since seems like other questionable corporate things were going on:
  - he was asleep below deck
  - the third mate was in charge of the wheelhouse
  - the ship's radar was broken
- But is a well-studied case

Johnson et al., Examined 3 Kinds of Cues

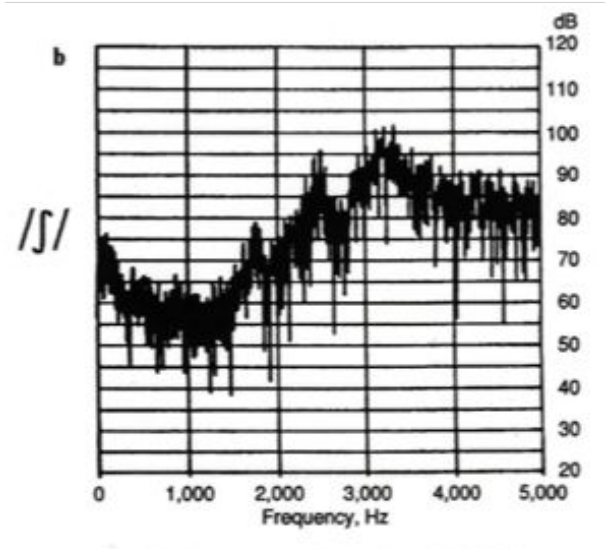
- Segmental Effects (phoneme, syllable, word level)
- Disfluencies
- Suprasegmental Effects (stress, intonation, etc.)

Keith Johnson's /s/ and /ʃ/

- Produced in a quiet recording booth with equipment responsive to up to 5,000Hz



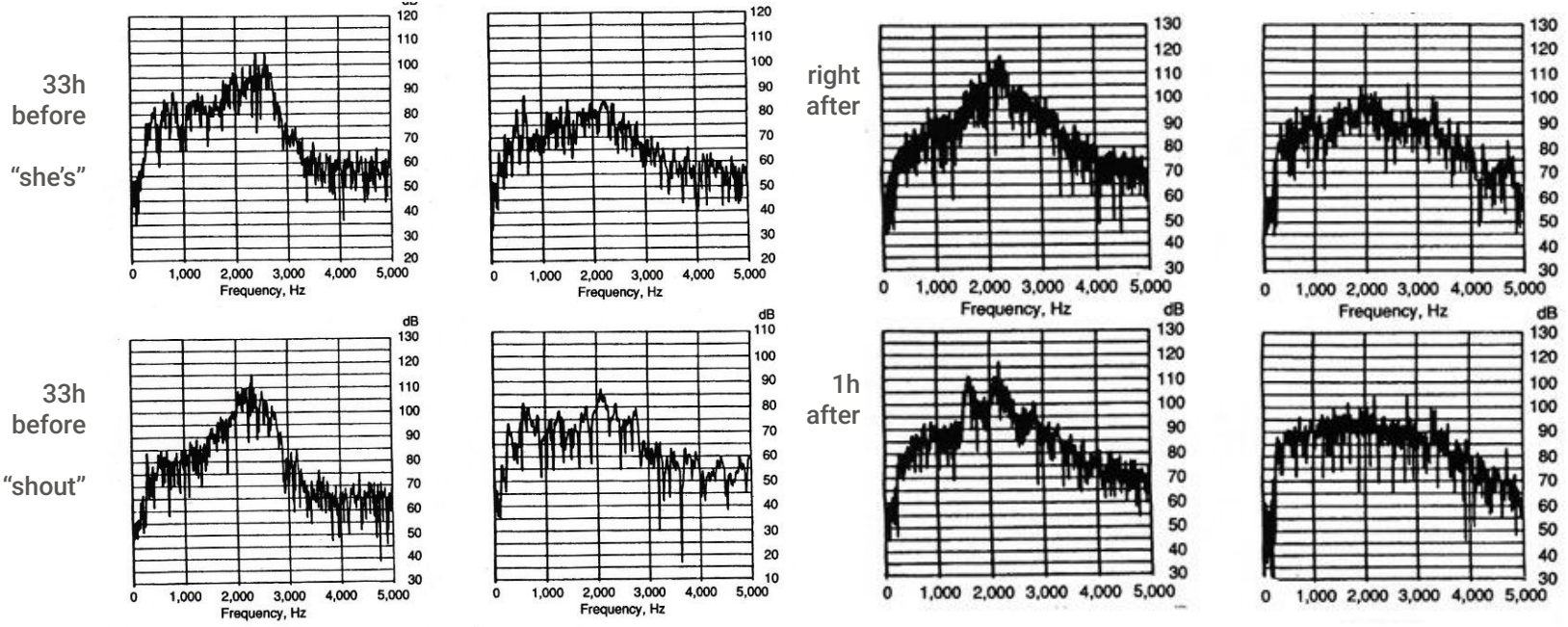
/s/ - as in “sun”



/ʃ/ - as in “shun”

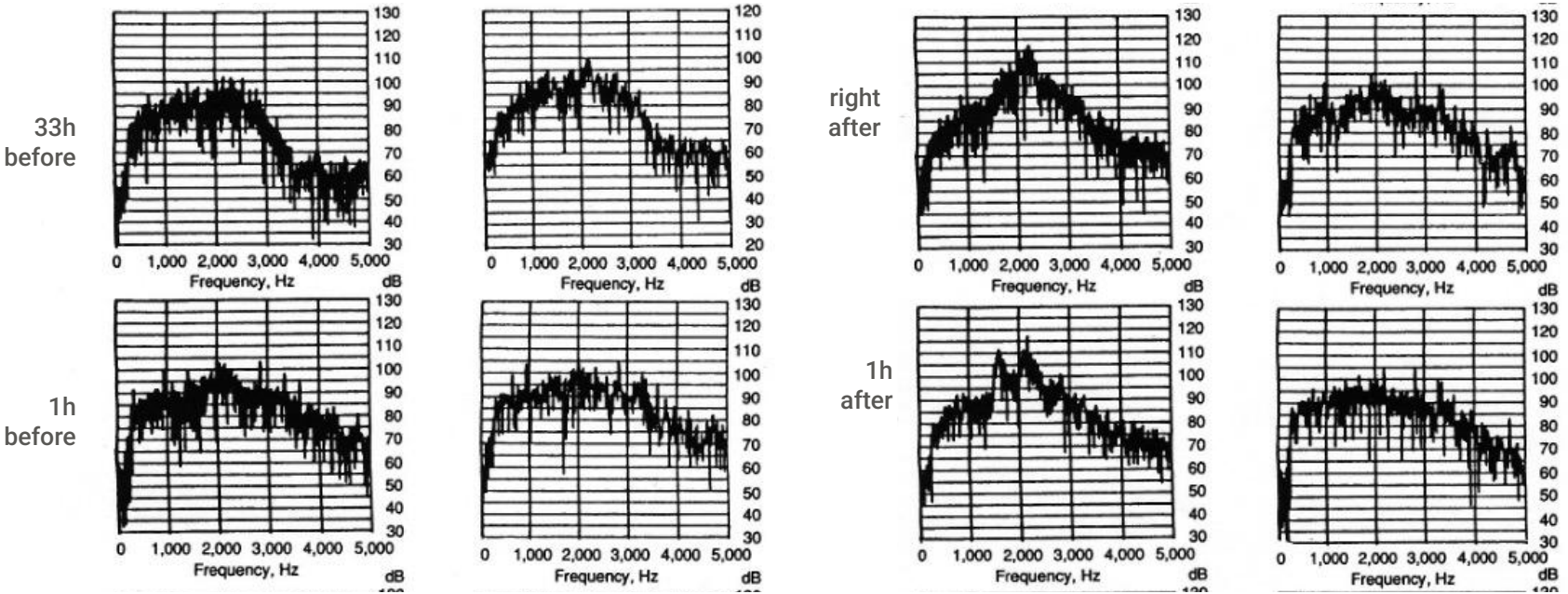
Captain Hazelwood's /ʃ/

- Power of spectra /s/ paired with spectra of nearby open mike pauses from NTSB recordings 33 hours before, immediately after and 1 hour after the accident.

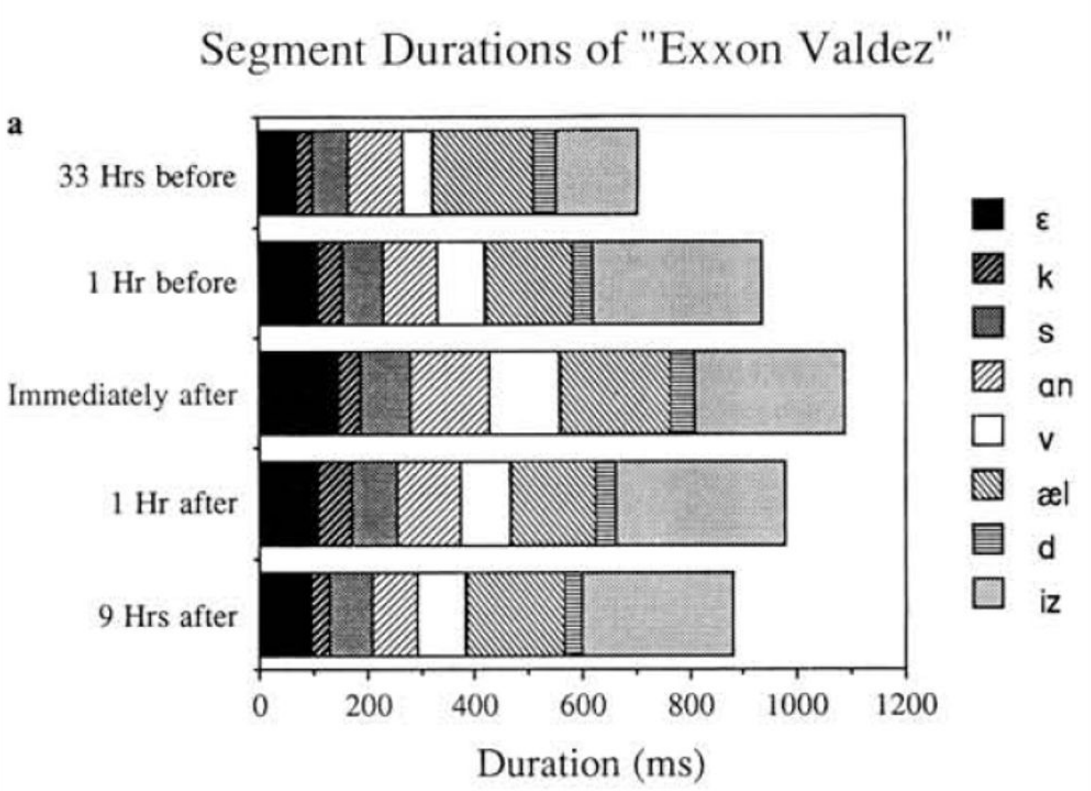


Captain Hazelwood's /s/

- Power of spectra /s/ paired with spectra of nearby open mike pauses from NTSB recordings



Duration


Segment Durations of "Exxon Valdez"
a
33 Hrs before
1 Hr before
Immediately after
1 Hr after
9 Hrs after
0
200
400
600
800
1000
1200
Duration (ms)
ɛ
k
s
ɑn
v
æl
d
iz

Summary

Gross effects	revisions (–1) Exxon Ba, uh Exxon Valdez (–1) departed disembarked (–1) I, we'll (–1) columbia gla, columbia bay
Segmental effects	misarticulation of /r/ and /l/ (0) northerly, little, drizzle, visibility (/s/ becomes /ʃ/ (fig. 3) final devoicing (e.g. /z/ → /s/) (–1,0,+1) Valdez → Valdes
Suprasegmental effects	reduced speaking rate (fig. 4, 5) mean change in pitch range (talker-dependent, fig. 6) increased F_0 jitter (fig. 6)

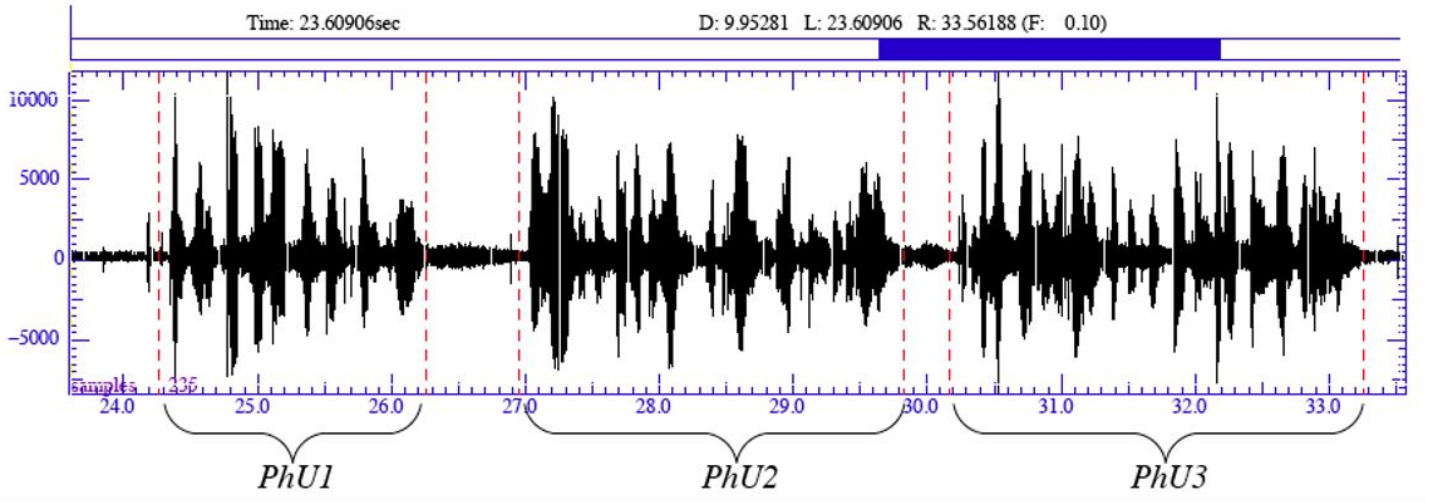
Table: Summary of phenomena found in the analysis of the NTSB tape (numbers in parentheses indicate the time of recording)

Problems

- If intoxicated speech, why wasn't s pronounced as "sh" 1 hour before?
- Other kinds of speaker state could cause drop in F0, slower speech, and disfluencies?
- Stress, just having woken up, trauma….

Automatic Classification

- **Use of prosodic speech characteristics for automated detection of alcohol intoxication**
  - Michael Levit, Richard Huber, Anton Batliner, Elmar Noeth
- **Break utterance into phrases automatically, based on**
  - fundamental frequency (where possible);
  - zero-crossing rate



Then Use 4 Classes of Features

- Prosodic
  - F0 max, F0 min, energy max, energy min, pause length
- Duration of voiced regions, unvoiced regions, etc.
- Jitter and shimmer
  - jitter is variation in pitch
  - shimmer is variation in energy
- Average cepstrum and cepstral slope

Methods

- Alcoholized speech samples collected at the Police Academy of Hessen, Germany
  - 120 readings (87 minutes) of a fable
  - 33 male speakers
  - BAC between 0 and .24/mille

Alcohol Blood Level	0.0	< 0.4	< 0.8	< 1.2	< 1.6	< 2.0	< 2.4
Recordings	32	20	20	18	20	7	3

  - Binary task: above or below 0.8/mille
  - leave-one-out cross-validation
  - neural net classifier

Levit et al., Results

- Used dev set to find best classifier
- This suggested two feature classes:
  - Prosodic features
  - Jitter/shimmer
- Results with this classifier
  - 62% phrase-accuracy
  - 69% for the whole speech sample
    - voting of the phrases

Alcohol Language Corpus

Florian Schiel et al 2009, 2010

- Bavarian Archive for Speech Signals
  - 162 speakers (77 female, 85 male)
  - recorded in a car (sometimes with engine running)
  - command and control speech (“turn off the radio”)
  - spontaneous dialogue, monologue, question answering
  - read speech
  - counts of disfluencies, etc

Sample drunk

Sample sober

Automatic detection in ALC: Paralinguistic Challenge 2011

- **Human**: 66-72% (Schiel 2011, Ultes, Schmitt, Minker 2011)
- **Machine**: roughly 65%-70%
- Example features from winning system:

  Bone, Daniel, Matthew Black, Ming Li, Angeliki Metallinou, Sungbok Lee, and Shrikanth S. Narayanan. 2011.
  Intoxicated Speech Detection by Fusion of Speaker Normalized Hierarchical Features and GMM Supervectors. In INTERSPEECH, pp. 3217-3220.

  - Prosody (f0, duration, energy, jitter, shimmer)
  - Spectral (MFCC, MFB log-energy, formants)
  - Computed over whole utterance and small windows
  - normalized phoneme duration
  - iterative speaker normalization