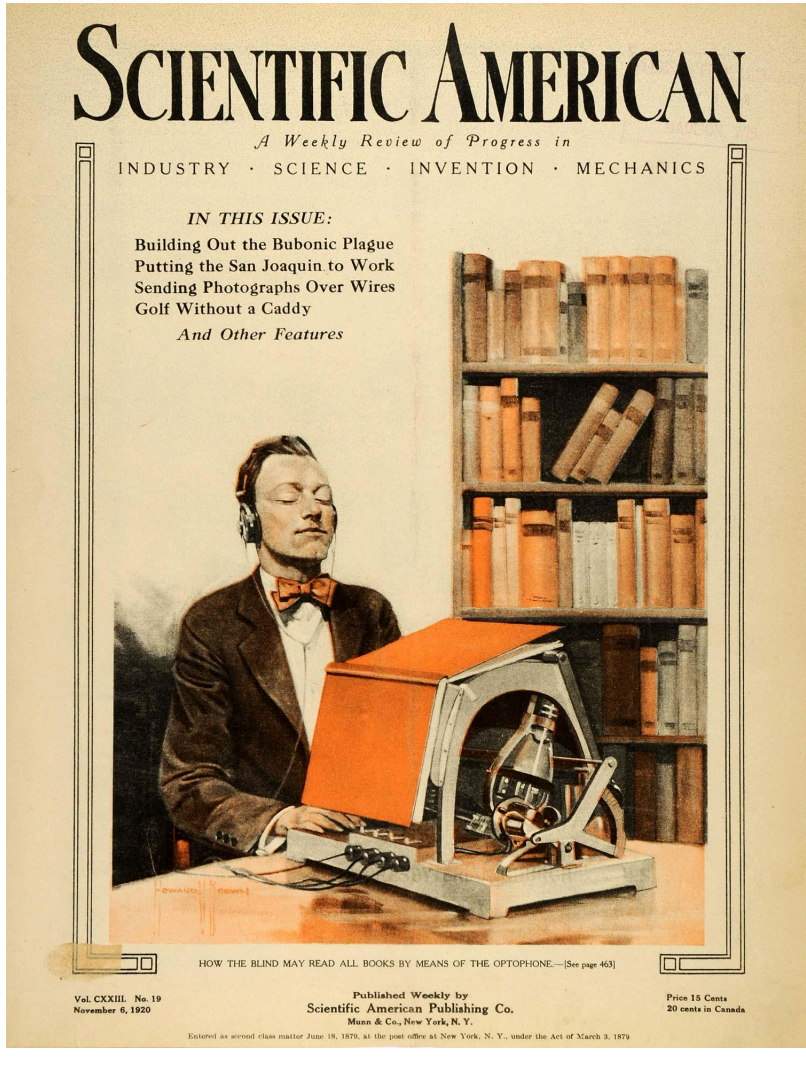
OCR & Transcription

Session 04 · Forschungspraktikum CSS

Dr. Sascha Grehl

2026-11-12



Outline

- Optical Character Recognition
  - Brief history
  - Problems and pitfalls
  - Tesseract
- Automated transcription
  - Transcription in sociology
  - Challenges
  - OpenAI Whisper
  - Diarization
- Where are we right now?

What Are We Talking About Today?

- Last week: how to acquire digital trace data
  - Usually in text format
- Today: data that come in a different shape and need to be transformed
  - PDFs or images containing text
  - Interviews and videos containing spoken text

End goal: text

Optical Character Recognition (OCR)

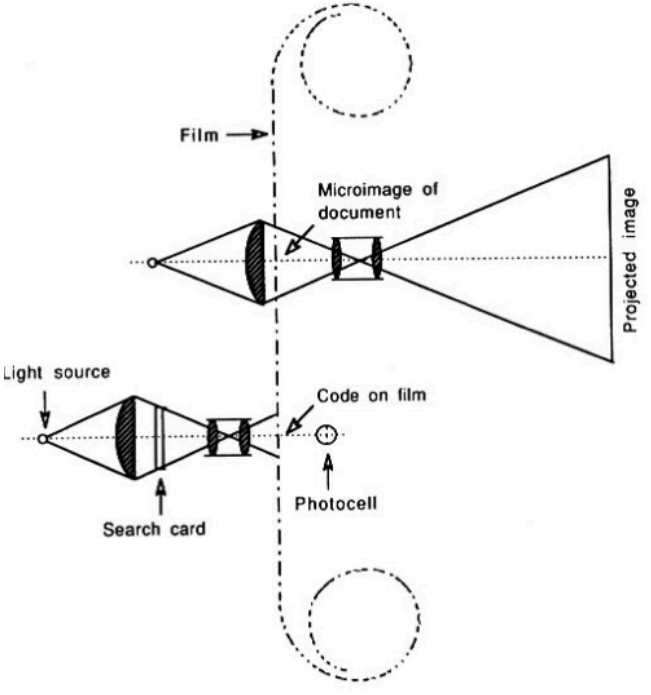
Optical Character Recognition

- Some documents might not be machine-encoded yet
  - Machine-encoded: text that can be edited, searched, or displayed online
- Examples:
  - PDFs downloaded from newspaper archives
  - Book scans, including old books
  - Academic papers for “science of science” projects

Emanuel Goldberg

Early pioneer: **Emanuel Goldberg**

- “Statistical Machine” for document retrieval
- Letters were punched into a search card
- Light was sent through the card to a photocell
- When the pattern matched a pattern in the film, no light reached the photocell and an alarm was triggered

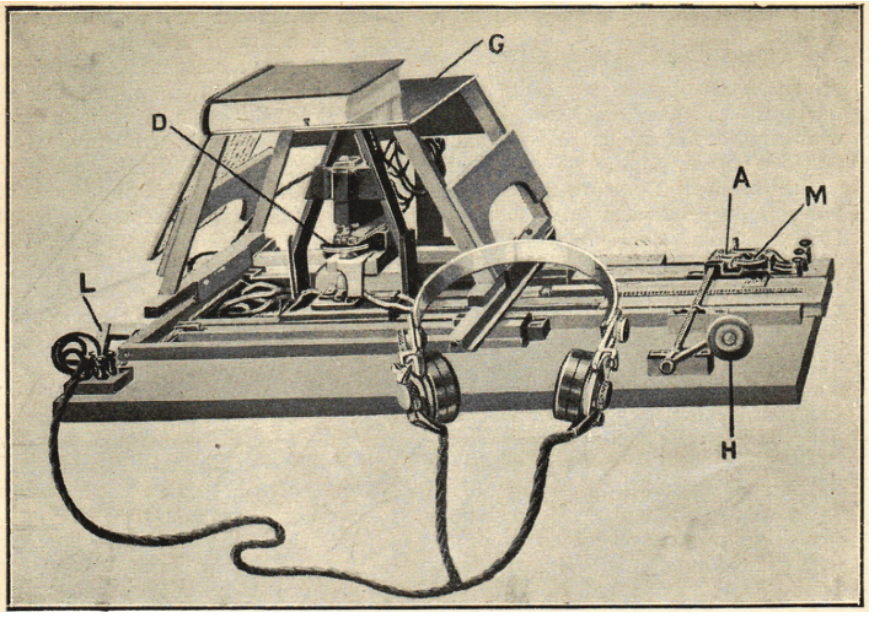


Goldberg’s Statistical Machine

The Optophone

Early pioneer: **Edmund Fournier d'Albe**

- Device for blind people
- Scans a page and produces different sounds based on the letters it "sees"



Fournier d'Albe's Optophone

Omni-Font OCR

Early devices worked only with particular fonts

1974: Ray Kurzweil

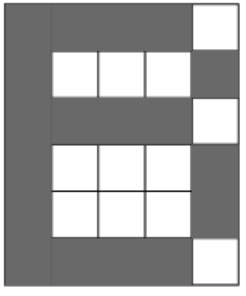
- “Omni-font OCR”
- A scanner scanned the text
- Characters were mapped to sounds to read text aloud



Ray Kurzweil and the Reading Machine

Matrix Matching (Historical Approach)

Input sample character



Template characters in template database

		1		
	1	1	1	
	1		1	
1	1	1	1	1
1				1
1				1

A

1	1	1	1	
1				1
1	1	1	1	
1				1
1				1
1	1	1	1	

B

	1	1	1	
1				1
1				
1				
1				1
	1	1	1	

C

Digitized character

1	1	1	1	0
1	0	0	0	1
1	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	1	1	1	0

Digitized sample compared with stored templates

X	X	1	X	
X	X	X	X	X
X	1	X	1	
1	X	X	X	1
1				1
1	X	X	X	X

17 mismatched points

1	1	1	1	
1				1
1	1	1	1	
1				1
1				1
1	1	1	1	

0 mismatched points

X	1	1	1	
1				1
1	X	X	X	
1				X
1				1
X	1	1	1	

6 mismatched points

Character Recognition by Matrix Matching

OCR: Problems and Pitfalls

Text documents come often in an suboptimal format.

Problem	Strategy
Tilted scans	Deskewing
Speckles, black dots, and other noise	Despeckling
Text in different colors	Binarization
Words and characters run together	Segmentation

We will use `magick` for image preprocessing

Example



Skewed Document Before OCR (Source: https://www.klippa.com/en/blog/information/what-is-ocr/)

Tesseract

- We will use Tesseract
- Originally developed by HP as part of its scanner work
- Open-sourced in 2005
- Maintained by Google from 2005 to 2018
- **Important: Must be installed as a system application first**
- Only then can it be used from R

Tesseract and Preprocessing

- Tesseract requires well-preprocessed images for accurate OCR:
  - Resolution: Scale up images so text is sufficiently large (x-height ≥ 20 px)
  - Alignment: Correct rotation and skew
  - Brightness: Remove uneven lighting and background variations
  - Borders: Remove dark borders to prevent character misidentification
- Again, see package: `magick`

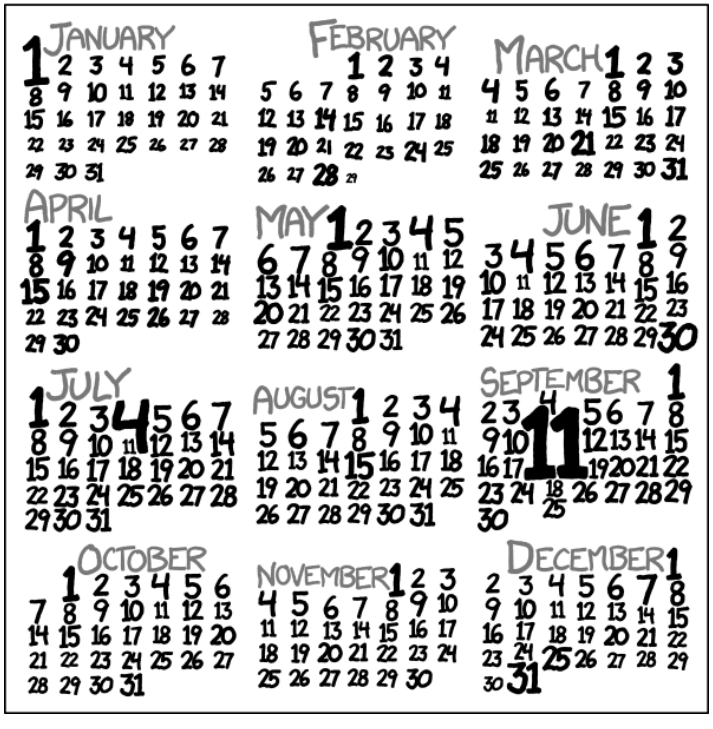
Tesseract Example

RStudio IDE (or **RStudio**) is an integrated development environment for R, a programming language for statistical computing and graphics. It is available in two formats: RStudio Desktop is a regular desktop application while RStudio Server runs on a remote server and allows accessing RStudio using a web browser. The RStudio IDE is a product of Posit PBC (formerly RStudio PBC, formerly RStudio Inc.).

An example text from Wikipedia

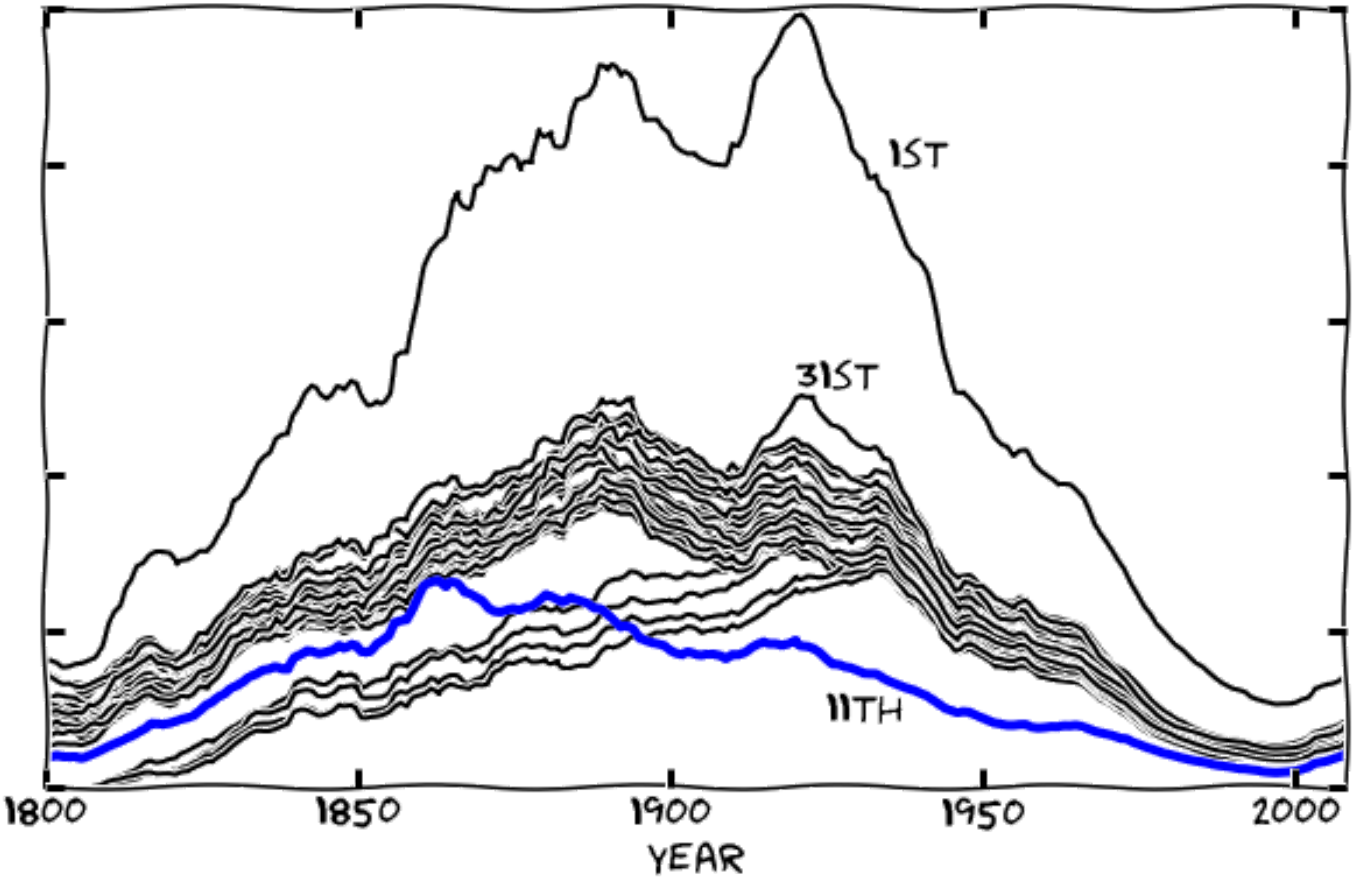
```
# A tibble: 10 × 3
   word        confidence bbox
   <chr>            <dbl> <chr>
 1 RStudio           92.1 14,18,134,43
 2 IDE               96.9 145,19,196,42
 3 (or               93.3 207,18,245,49
 4 RStudio)          92.3 256,18,385,49
 5 is                96.4 397,19,418,43
 6 an                96.4 429,24,461,43
 7 integrated        96.4 474,18,612,49
 8 development       93.7 624,18,806,49
 9 environment       96.4 816,19,991,43
10 for               93.3 1001,18,1038,43
```

Up to 11



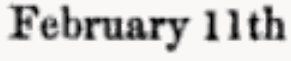
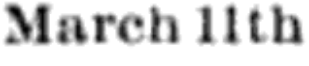
"In months other than September, the 11th is mentioned substantially less often than any other date. It's been that way since long before 9/11 and I have no idea why." (https://xkcd.com/1140/)

David R. Hagens Enquiry

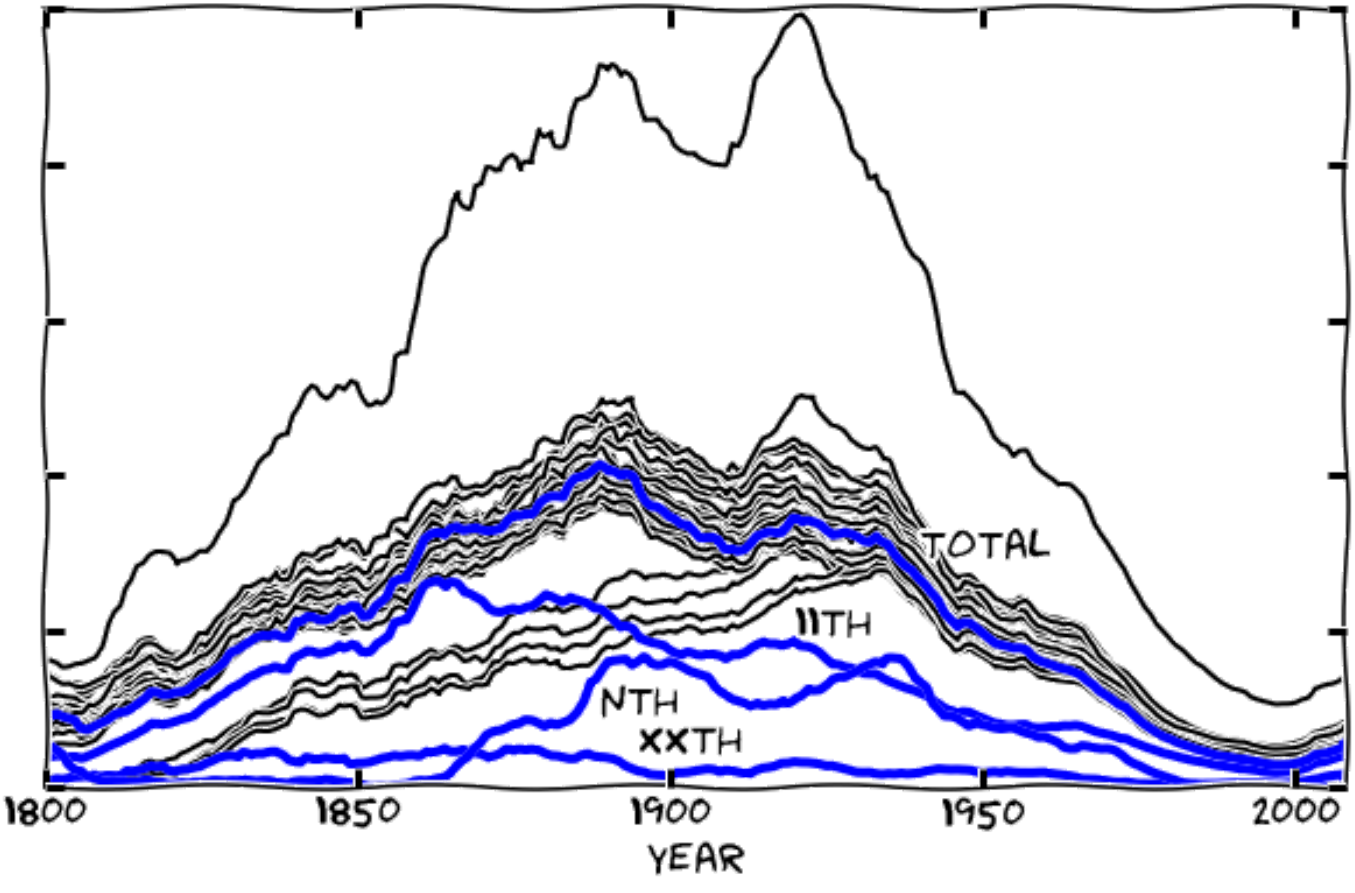


(https://drhagen.com/blog/the-missing-11th-of-the-month/)

Typographical hijinks

Image	Googles OCR
	`January IIth`
	`February llth`
	`March lith`
	`January nth`

David R. Hagens Enquiry II (or 11?)



(https://drhagen.com/blog/the-missing-11th-of-the-month/)

Post-Processing

Keep an eye out for potential systematic errors in your data

- Match words with lexicons
- Use neighboring words to correct errors
  - Example: `Washington DoC.` → `Washington D.C.`
- A pretrained language model can supply context

Transcription

Transcription

Spoken language is another common way of storing text

Examples:

- Interviews
- Speeches
- Podcasts
- Videos

Why not use a dedicated classifiers for audio data?

Why Transcribe?

- Computionational more demanding
- A unified format: **text**
- Easier annotation
- Less audio noise in subsequent text analysis
- Easier interpretation

> “[Transcription] renders the big booming confusion of raw conversation into forms that support inquiry” – Vanover 2022: 64

When I Was Your Age… I Took a Course on Qualitative Sociology

- Transcription by hand is the classic approach in qualitative sociology
- Assumption: transcribing helps researchers immerse themselves in the data

> "transcription slow[s] the process down and create[s] time for reflexivity and theory-building" – Vanover 2022: 64

- Research and interpretation happen while transcribing
- Strategies include:
  - Word-by-word transcription
  - Naturalized transcription: adding context and representing how something was said

Atheoretical Transcription: Example

Version 1: An Atheoretical Transcription

And the reading special came in and just said you can't be in here. You have to go to your class. And that it was like she started sort of getting real direct with him. Telling him no, you are going to your class. You cannot stay in here. And I said and I sort of pulled her aside and I said you know his dad just passed away this morning. He's really calm. I don't mind him being in here. And she was like No, you are being very nice. But that's ridiculous. He has to go to his other class. Whipped him into a frenzy. Just you're going to this class. I'm not going to that class. You're going to this class. I'm not going to that class I mean just really and I stepped back because honestly I did not know what to do.

Atheoretical Transcription of an Interview

Naturalized Transcription: Example

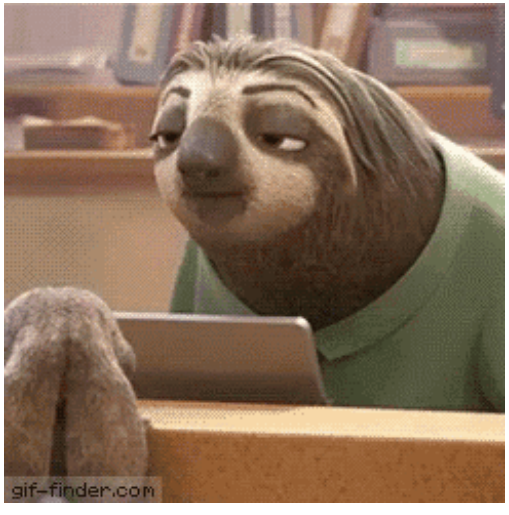
Version 2: A Narrativized Transcription of the Same Interview Segment

And the reading special came in and just said
You can't be in here. You have to go to your class.
And that, it was like, she started sort of getting real direct with him. Telling him,
No, you are going to your class. You cannot stay in here.
And I said—and I sort of pulled her aside and I said
You know, his dad just passed away this morning. He's really calm. I don't mind him being in here.
And she was like
No, you are being very nice. But that's ridiculous. He has to go to his other class
Whipped him into a frenzy. Just
You're going to this class.
I'm not going to that class
You're going to this class
I'm not going to that class
I mean, just really. And I stepped back because honestly I did not know what to do.

Naturalized Transcription of the Same Interview

The Cost of Manual Transcription

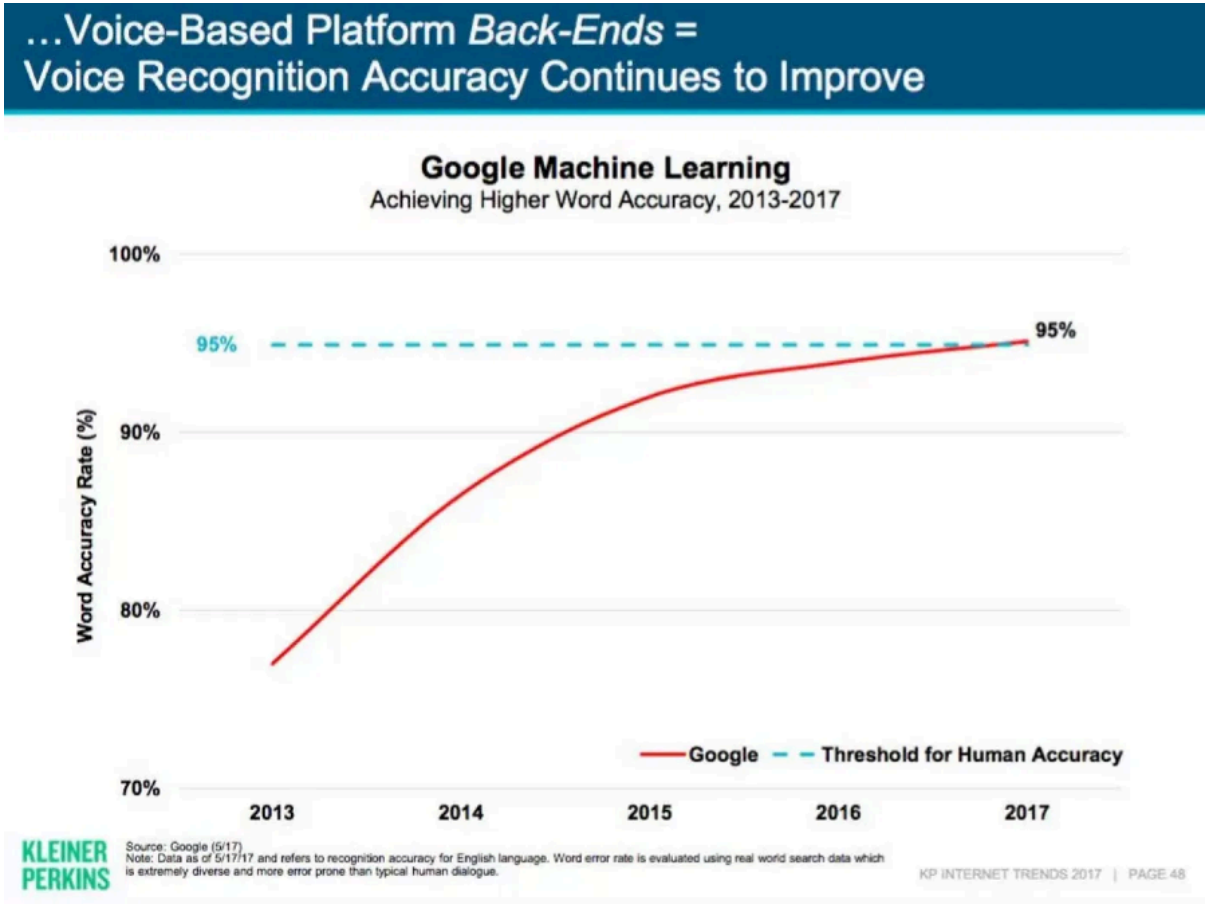
- Manual transcription is time-consuming



"Only 59 minutes of the interview left."

- This limits its use for large collections of recordings
- Automated transcription can help process larger samples

Speech Recognition: Historical Progress



Historical Improvements in Speech Recognition Accuracy

How Transcription Works

- Audio is continuous
- Sampling records observations at a given rate, e.g. 16,000/s
- Features are extracted from the signal
- Phonemes are basic units of sound:
  - “th” in “think”: /θ/
  - “ee” in “see”: /iː/
- Modern speech recognition predicts words (positional encoding)
- Post-processing handles grammar, punctuation, and spelling



Waveforms of Tonal Sound and Noise

Speech Recognition as Prediction

- A simplified prediction task: words from speech sounds: words ~ phonemes
- Models learn from paired speech and text
- Problem: Language is very diverse
- Solution: A **really large** model trained on many examples

–> We will use **OpenAI Whisper**

Whisper

- Trained on paired speech and text
  - 680,000 hours of labeled internet speech
  - 563,000 hours English, 117,000 hours in 96 languages
- Goal: Zero-shot model with highest possible accuracy
- **Zero-shot:** apply the pretrained model without fine-tuning on the target dataset

Whisper: Loading a Model

We can either use it in `R`

```r
install.packages("whisper")  # Install the R package
torch::install_torch()       # Install the deep learning backend (once)

library(whisper)             # Load Whisper
```

Or in `python`

```python
import whisper

whisper_model = whisper.load_model("base")

transcribe_audio("assets/session_04/MLKDream_64kb.mp3")
```

Whisper: Select a Model

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39 M	`tiny.en`	`tiny`	~1 GB	~10x
base	74 M	`base.en`	`base`	~1 GB	~7x
small	244 M	`small.en`	`small`	~2 GB	~4x
medium	769 M	`medium.en`	`medium`	~5 GB	~2x
large	1550 M	N/A	`large`	~10 GB	1x
turbo	809 M	N/A	`turbo`	~6 GB	~8x

```
1 download_whisper_model("base")
```

Whisper: Transcript Output

```
result <- transcribe(
    file = "MLKDream_64kb_shortened.mp3",
    model = "base",
    language = "en",
  )
result$text
```

```
[1] "I have the pleasure to present to you Dr. Martin Luther King. I am happy to join with
you today in what will go down in history as the greatest demonstration for freedom in the
history of our nation. Five score years ago, a great American in whose symbolic shadow we
stand today, signed the emancipation proclamation. This momentous decree came as a great
beacon light of hope to millions of Negro slaves who had been seared in the flames of
withering injustice. It came as a joyous daybreak to end the long night of their captivity.
But one hundred years later, the Negro Negro still is not free."
```

Whisper: Inspecting the Transcript

I have the pleasure to present to you Dr. Martin Luther King. I am happy to join with you today in what will go down in history as the greatest demonstration for freedom in the history of our nation. Five score years ago, a great American in whose symbolic shadow we stand today, signed the emancipation proclamation. This momentous decree came as a great beacon light of hope to millions of Negro slaves who had been seared in the flames of withering injustice. It came as a joyous daybreak to end the long night of their captivity. But one hundred years later, the Negro Negro still is not free.

Diarization

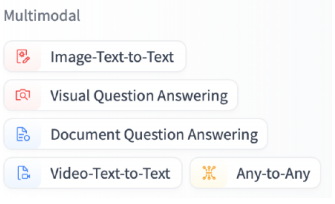
- Audio may contain several speakers
  - Interviews, talk shows, and discussions
- We want to distinguish speakers and attribute speech to them
- This process is called **diarization**
- Python's `pyannote.audio` package

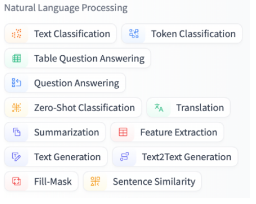
Diarization: Example

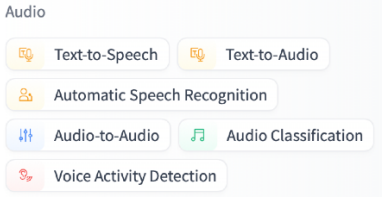


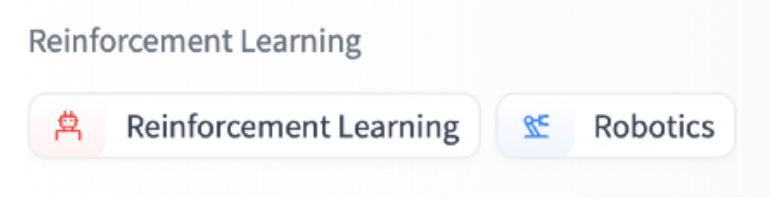
Pretrained Diarization Models

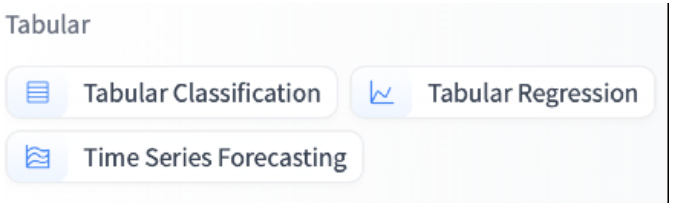
- `pyannote.audio` uses pretrained models
- The source illustrates downloading models from **Hugging Face**

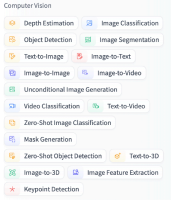


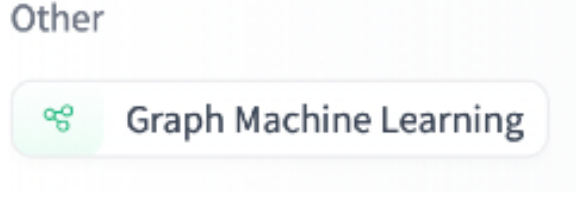






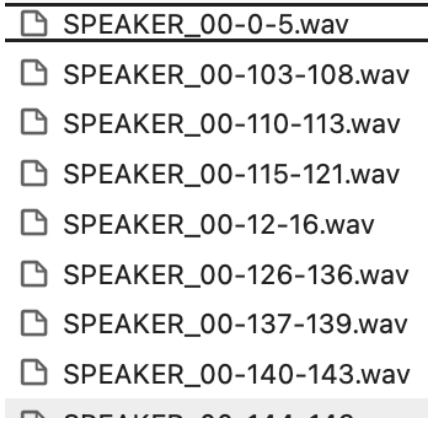






Diarization and Transcription Pipeline

1. Read the audio file
2. Identify speakers and their speaking segments with pyannote
3. Transcribe the segments with Whisper
4. Combine results into a table with **speaker**, **text**, **start**, and **end**



Audio Segments Grouped by Speaker

```
[{'speaker': 'SPEAKER_00',
  'start': 345.0,
  'end': 349.0,
  'text': ' 1963 is not an end, but a beginning.'},
 {'speaker': 'SPEAKER_00',
  'start': 137.0,
  'end': 139.0,
  'text': " so we've come here today."},
 {'speaker': 'SPEAKER_00',
  'start': 337.0,
  'end': 338.0,
  'text': ' will not pass.'},
 {'speaker': 'SPEAKER_00',
  'start': 623.0,
  'end': 624.0,
  'text': ' is redemption.'},
```

Transcript Segments with Speaker Labels and Timestamps

Computing Resources

- These workflows can require substantial computing resources, including GPUs
- The practical material covers Python and scientific computing basics:
  - Modules, environments, and scripts
  - Slurm jobs

These tools are available if your project needs them

Where Are We Right Now?

- How is it going?
- Which data acquisition topics need more explanation?
- Has the material been useful so far?
- Is the pace manageable?

Merci!

sascha.grehl@uni-leipzig.de

Universität Leipzig