SPEECH RECOGNIZER

KSENIYA BOUT

NICK HALLIWELL

AINA LOPEZ

YAROSLAV MARCHUK

INDEX

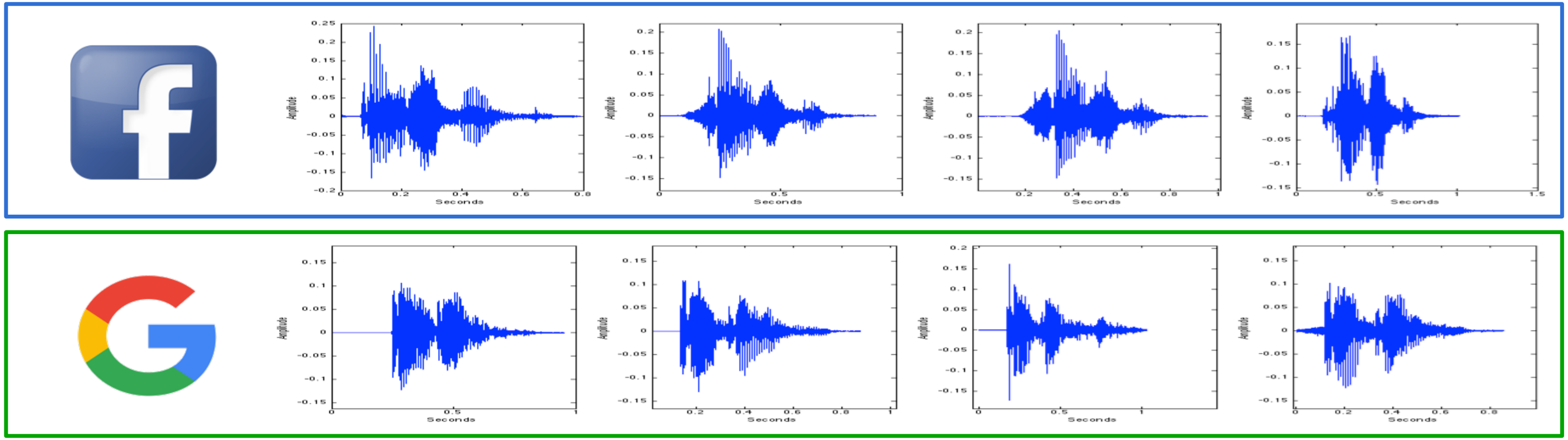
1. **Introduction**
2. **Data and Speech Recognition**
3. **Dynamic Time Warping (DTW)**
    a. Algorithm
    b. Conditions
4. **Speech Recognizer**
5. **Conclusions**

INTRODUCTION

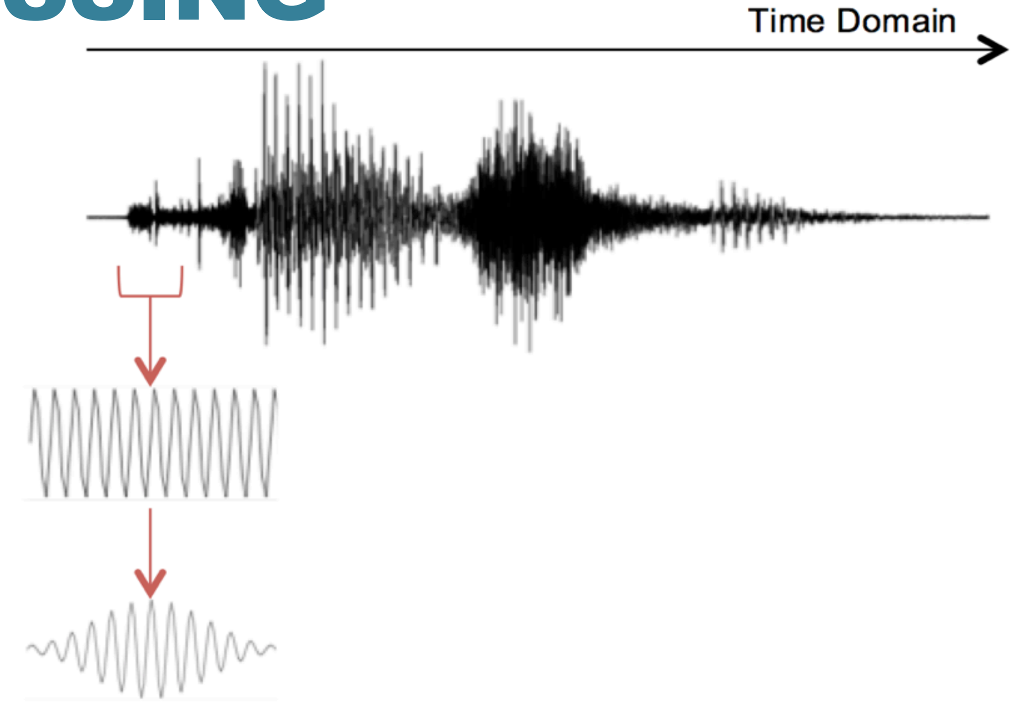
- We are interested in recognizing speech (words “Google” and “Facebook”) .
- We find that it is done by Dynamic Time Warping algorithm (Dynamic Programming).
- We learn its theory and applications.
- We implement this all in R. We record words, we extract features, we write DTW algorithm, we obtain successful results.
- We enjoy the speech recognition!

DATA

- We have sound files with the words *Google* and *Facebook*.
- Sound Waves are **non-stationary**.
- We can’t compare non-stationary signals (directly).
- Solution? → Apply **speech processing** techniques



SPEECH PROCESSING



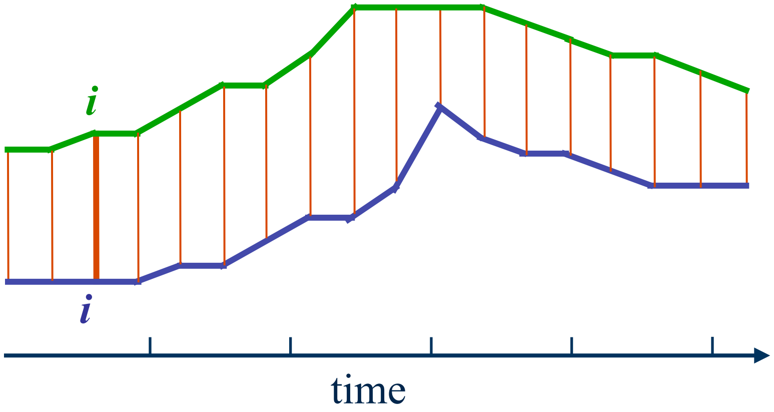
1. **Pre-emphasize:** Boost the energy of higher frequency components

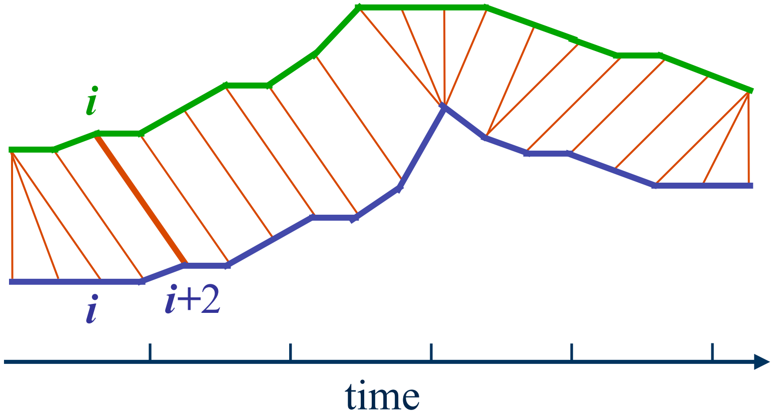
2. **Framing:** assume that a small sample (16 ms) is stationary.

3. **Windowing:** Smoother transition between frames.

4. **Mel Frequency Cepstral Coefficients:** based on human perception of frequencies. Only keep the most relevant frequencies.

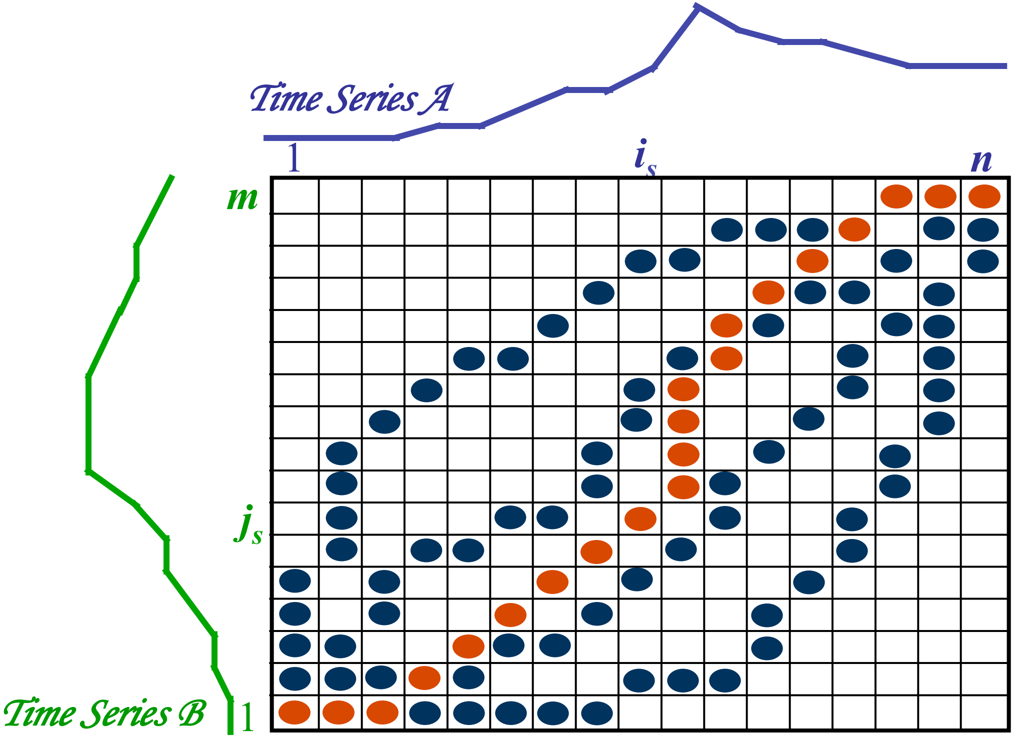
DYNAMIC TIME WARPING: INTUITION





* Images From: http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm

DYNAMIC TIME WARPING: ALGORITHM



There are a lot of possible warping paths through the grid.

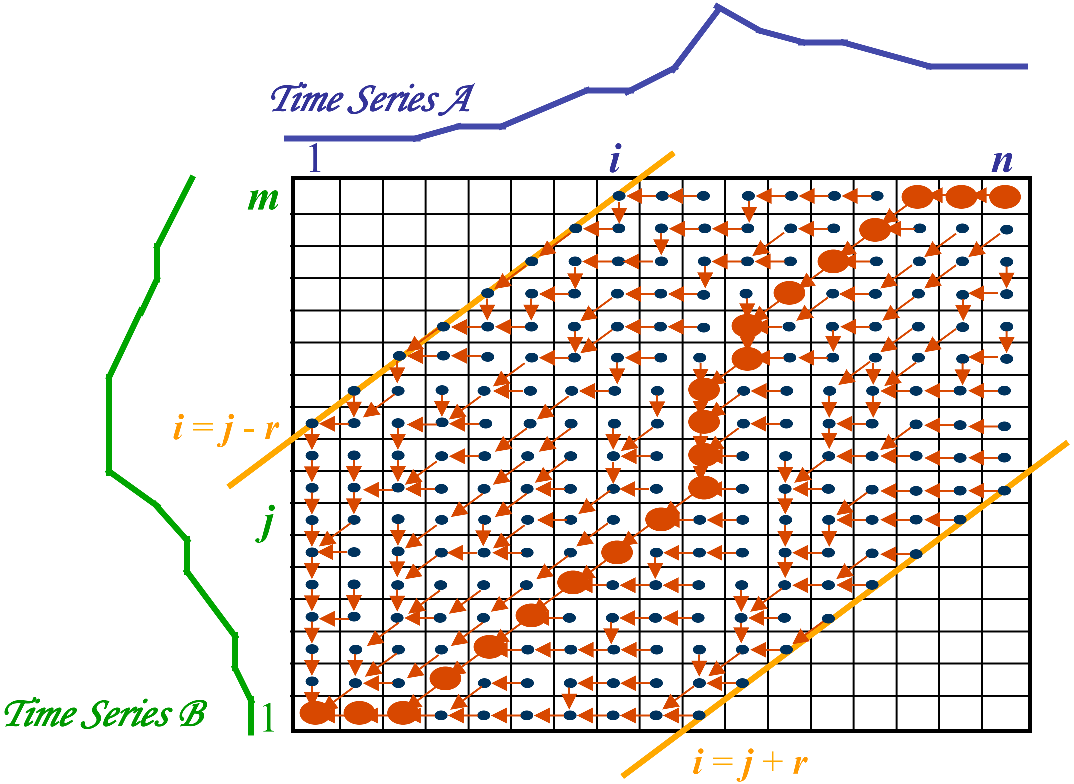
reduction of the search space

Restrictions on the warping function:

- monotonicity
- continuity
- boundary conditions
- warping window
- slope constraint.

* Images From: http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm

DYNAMIC TIME WARPING: ALGORITHM

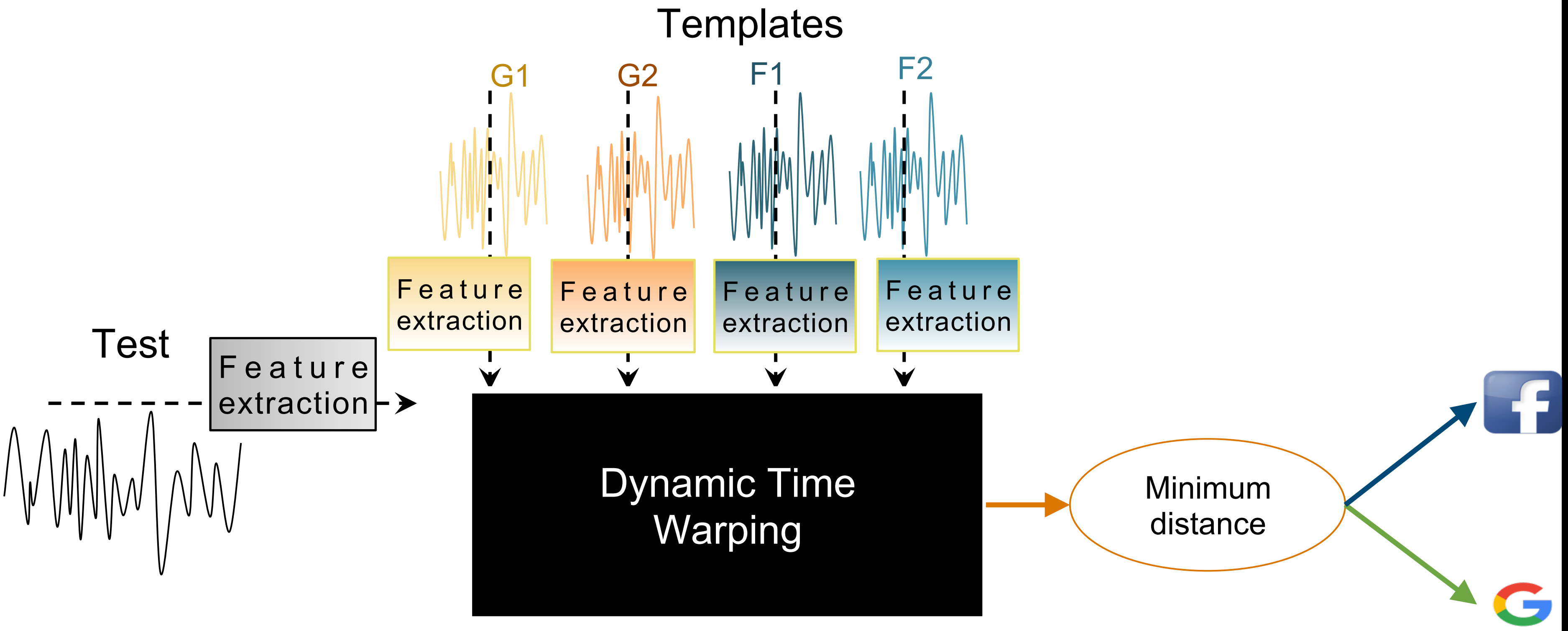


DP-equation:

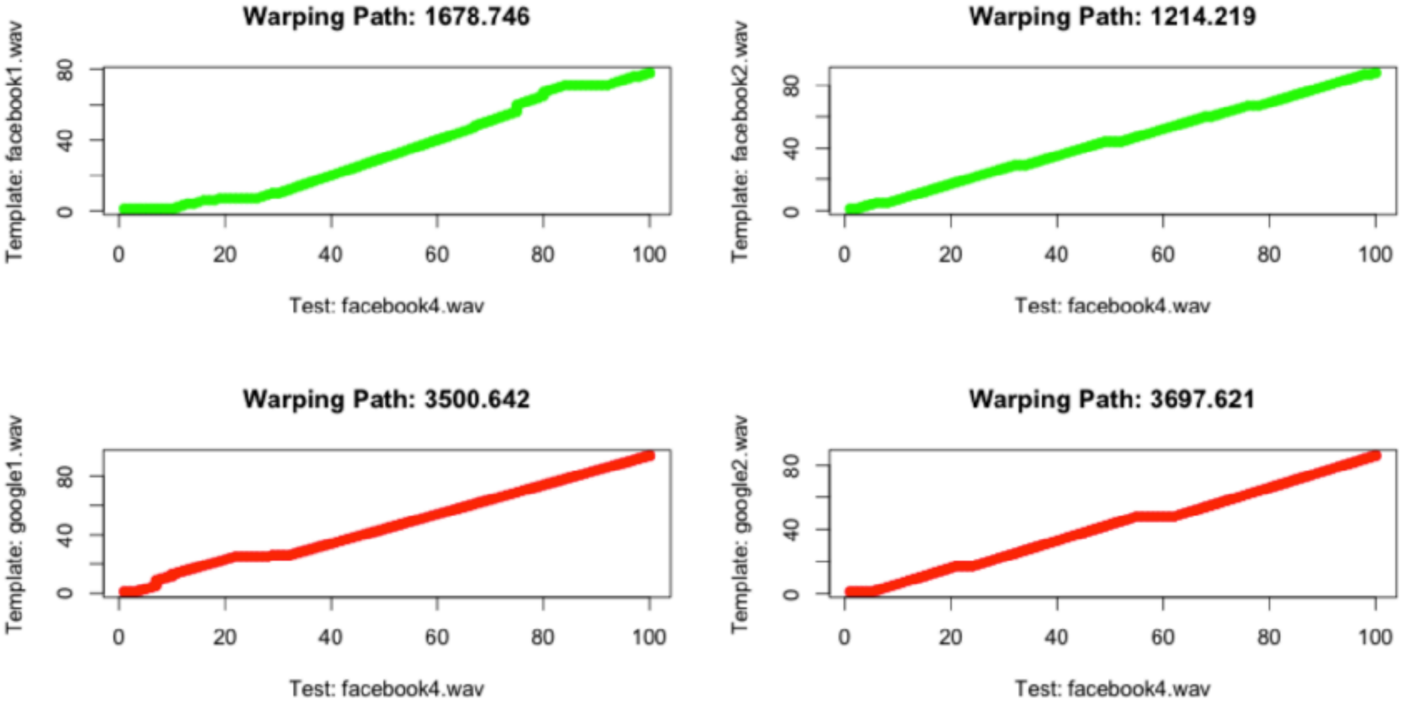
$$g(i, j) = \min \begin{cases} g(i, j-1) + d(i, j) \\ g(i-1, j-1) + 2d(i, j) \\ g(i-1, j) + d(i, j) \end{cases}$$

* Images From: http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm

SPEECH/WORD RECOGNIZER

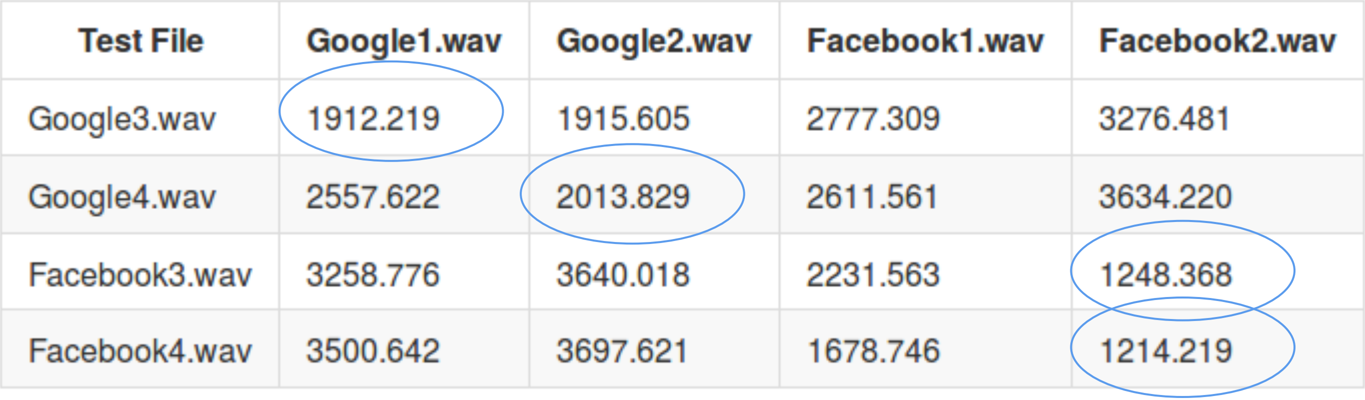


RESULTS


Warping Path: 1678.746
Template: facebook1.wav
Test: facebook4.wav
0
40
80
0
20
40
60
80
100
Warping Path: 1214.219
Template: facebook2.wav
Test: facebook4.wav
0
40
80
0
20
40
60
80
100
Warping Path: 3500.642
Template: google1.wav
Test: facebook4.wav
0
40
80
0
20
40
60
80
100
Warping Path: 3697.621
Template: google2.wav
Test: facebook4.wav
0
40
80
0
20
40
60
80
100

RESULTS

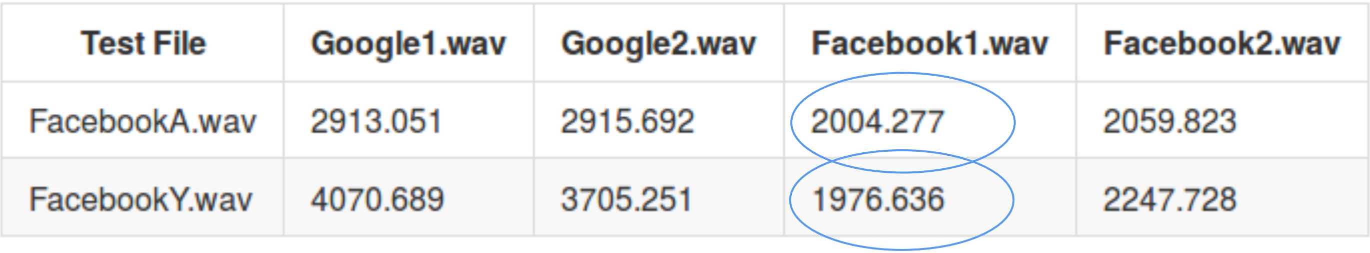
Same voice in test and templates



Test File	Google1.wav	Google2.wav	Facebook1.wav	Facebook2.wav
Google3.wav	1912.219	1915.605	2777.309	3276.481
Google4.wav	2557.622	2013.829	2611.561	3634.220
Facebook3.wav	3258.776	3640.018	2231.563	1248.368
Facebook4.wav	3500.642	3697.621	1678.746	1214.219

RESULTS

Different voices



Test File	Google1.wav	Google2.wav	Facebook1.wav	Facebook2.wav
FacebookA.wav	2913.051	2915.692	2004.277	2059.823
FacebookY.wav	4070.689	3705.251	1976.636	2247.728

CONCLUSIONS

- DTW is **fast** and **accurate**.
- The recognizer is **robust to gender** and works well on **individual words**
- We **don't have enough data** to make universal claims at this time.

- TO DO's:
  - More template data.
  - Test it with sentences.
  - From R, we were not able to import voices in real time, this would add a unique capability to our algorithm.

THANKS FOR YOUR ATTENTION!

QUESTIONS?