

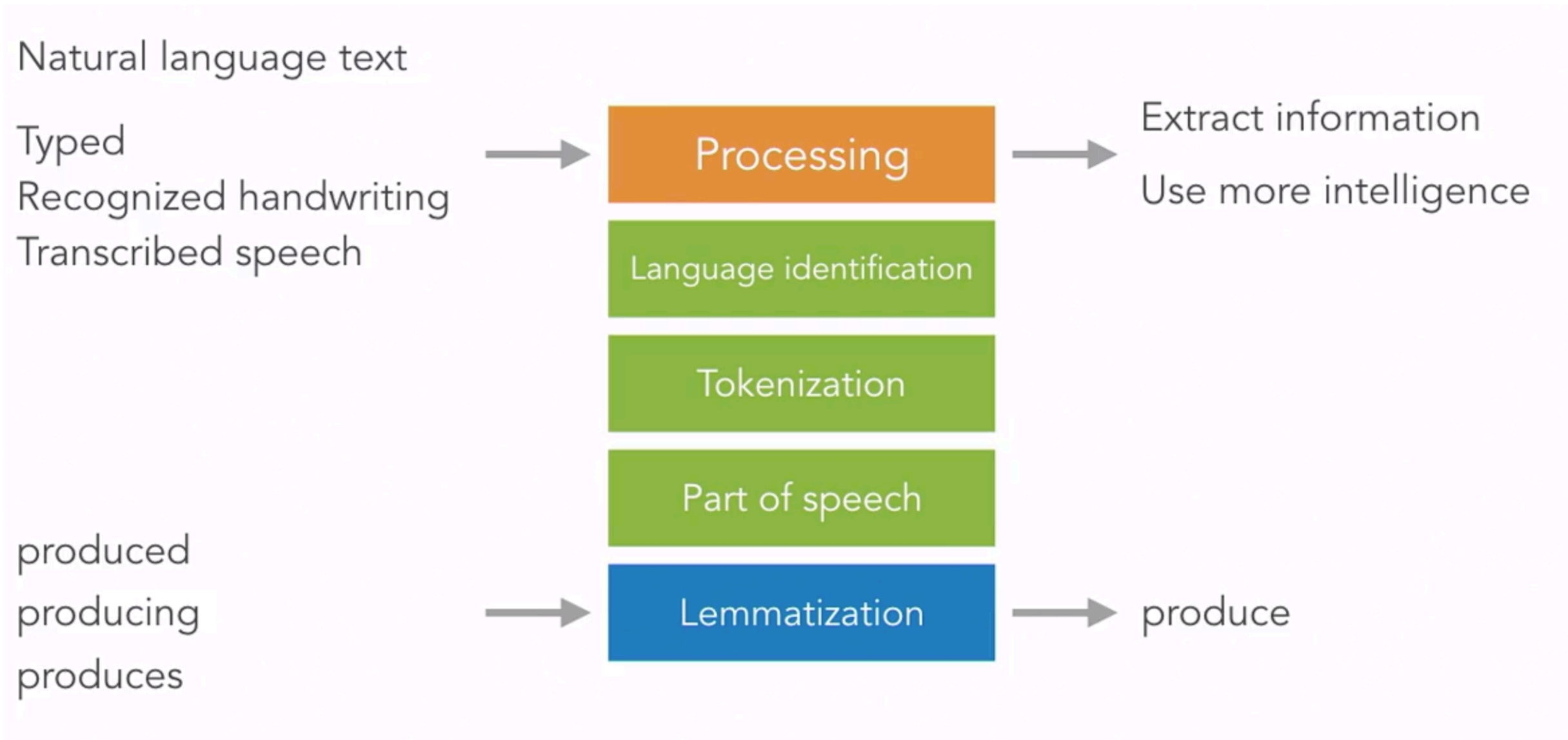
Adding Intelligence to Applications

Professor Larry Heimann
Carnegie Mellon University
Information Systems Program

In what ways can we make our mobile apps more “intelligent”?

What are the advantages of “intelligent” apps?

Natural language processing (NLP)



NLP and Lemmatization

- Lemmatization is a process of determining the lemma of a word based on its intended meaning.
 - e.g., the word “**better**” has “**good**” as its lemma
- Lemmatization is also the process of grouping together the inflected forms of a word so they can be analyzed as a single item.
 - e.g., **walk**, **walked**, **walks**, **walking**

Foundation has NLP built-in

(playground examples; NLP Diary)

Why would Apple bother building
NLP into iOS?

Mobile ML is many things



What is Core ML?

“CoreML is a machine learning framework that powers the iOS app developers to integrate machine learning technology into their apps.”

Why Core ML is great



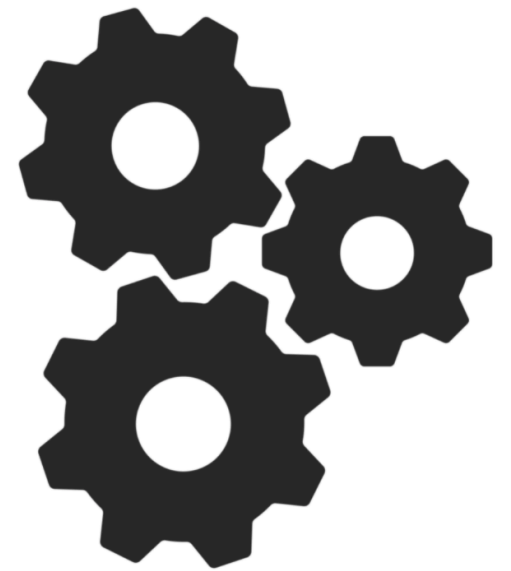
User Privacy



Server Costs



Always Available



Integrations

Building a Core ML model & app

(create_model.py; iPhone Pricer app)

Building a Core ML model with CreateML

(Sleep data analysis; cats and dogs)

Lab 9: Emoji Hunter

✓ Grade

✎ Edit Lab

✕ Delete Lab

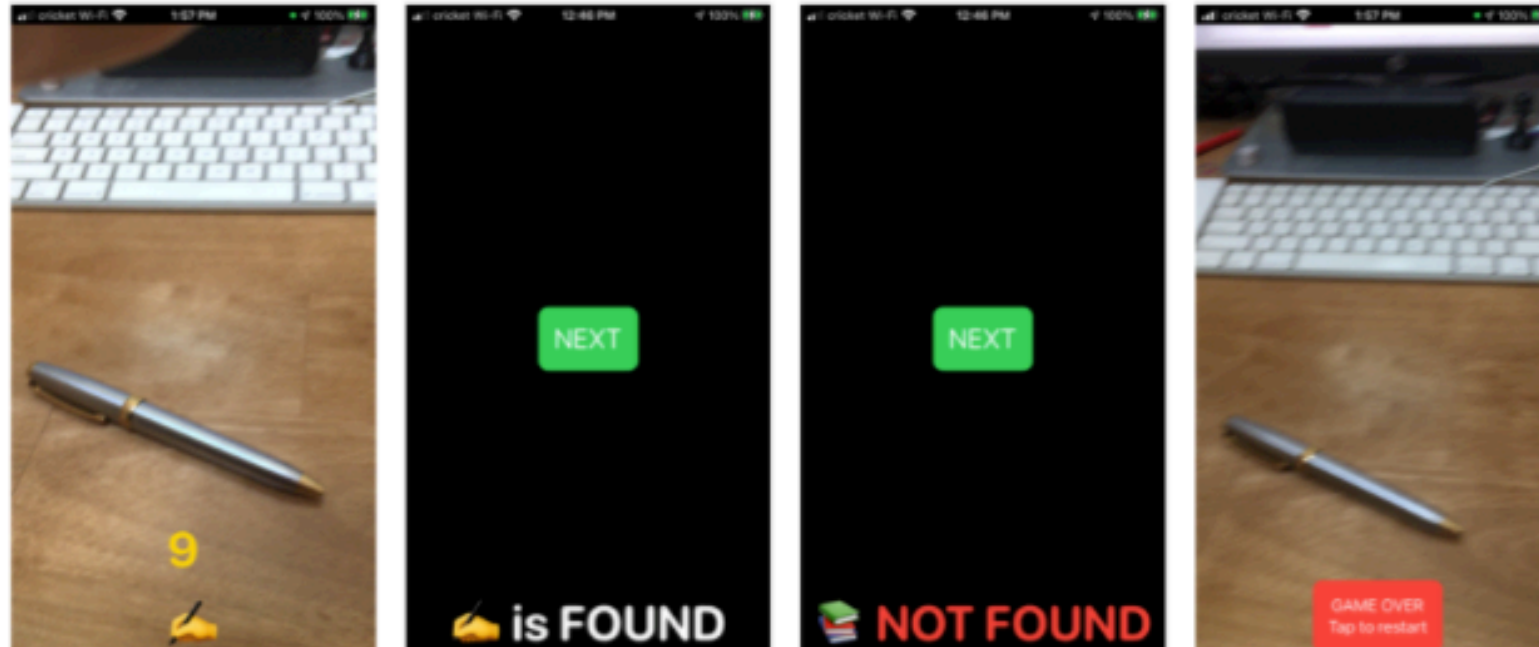
Due Date: November 20

Objectives

- reinforce key lessons on applied machine learning
- give students an opportunity to add CoreML into a functioning app
- allow students to interact with camera and the Swift Vision framework
- reinforce previous lessons on SwiftUI

📖 README.md

Our goal is to create a simple game called Emoji Hunter that will use our camera to identify real world object that look like emojis. A sample of the gameplay can be seen below:



When we start the Emoji Hunter game, our camera will be activated and the app will use CoreML to determine the dominant object in the frame; if it matches the target emoji within the specified time (10 seconds), then we have success.

Please note that because it uses the camera, this game has to be built on a device – the simulator is of limited value for running this app.

1. Begin by creating an app with SwiftUI (doesn't matter which `Life Cycle` you choose) called `EmojiHunter`. At this point, we know the drill.

Qapla'