TAKE-CV

Casey Kennington

January 15, 2015

Abstract

Description of the TAKE-CV data collection.

1 Summary

The goal of this data collection is to form a corpus of referring expressions (REs) with a referred object called the *target* and a potential object called the *landmark* that is used to make a relative RE (e.g., *take the redd cross next to the green t*).

2 Materials

- microphone for the participant to record the REs
- camera for the objects which will be fed to OpenCV
- two monitors, one for the wizard and one for the participant to show which object is the target and landmark
- a PC to run the OpenCV/python scripts
- a PC to record the audio (could be the same as the above PC)

3 Lab Layout

See Figure 1. The participant is seated near a table with the pento objects. A camera oriented so that it displays what the participant sees on the table (i.e., "up" in the video is the far side of the table from the perspective of the participant). A microphone is also near the participant to pick up speech and beeps.

4 Task

The overall goal (like the TAKE Apr13 data) is to collect as many episodes of REs as possible in the time the participant interacts with our system (i.e., the wizard). An episode begins when a small screen shows an overlay of the Pentomino objects on the table. One object is then visibly selected by a green outline around the object's contour (with an optional landmark object). The participant is then to produce a RE sufficient that the wizard can resolve the object.

We hope that it will be mostly self-annotating. That is, each episode will be easily segmented into a snapshot of the scene, the target and potential landmark.

5 Procedure

Before Task Participants are to be brought into the lab and seated in front of a table with 36 pento objects on it in a marked area of the table. After signing the form and assuring agreement, they are to be explained the task.

Figure 1: Layout: table with pento objects, participant sits in front of it in a seat, a camera is above the table oriented in the same direction as the participant's viewpoint.

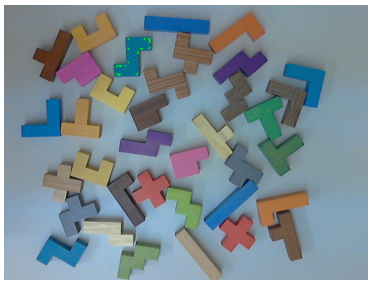

Figure 2: Example target only (in green).

Figure 3: Example target (in green) with landmark (in blue)

Figure 4: Example signal for reshuffling.

Explanation to Participant In front of you is a table with objects of various shapes and colors on it. The screen in front of you shows those same shapes. When you see a green outline highlighting a target object, you are to find that object on the table and identify that object such that a person sitting next to you would know what object you are talking about (Figure 2). You can identify objects in any way you want (e.g., use color, shape, side of the table, what object is near it, etc.). However, if you also see a blue highlighted object near the target object, we ask that you identify the target by using that as a reference point (e.g., you are next to the table). We ask that you only use speech to identify to the object, please don't point to or pick up the object. The system will attempt to determine which object you described. If it is successful, then a new target will be selected and the process will repeat. If it is not successful, a tone will sound, then it will select a new target and the process will repeat. We ask that you identify as many targets in the alloted time as possible, but please make sure the descriptions can uniquely identify the target.

At various times, a message "Please Shuffle" will appear on the screen (Figure 4). When that happens, please randomly move the objects around, but please be sure that all of the objects remain on the white area when you are done shuffling and that there are small gaps of space between the objects. After you are done shuffling, the message will disappear and a new target object will be selected as before.

After Task The participant is to be thanked and paid the agreed amount.

6 Software

We will be using OpenCV for object detection, using Livia Dia's python scripts to randomly select the target and landmark objects. OpenCV provides the ability to make multiple windows, so we can use that directly to create the wizard and the participant windows.

We will also record the audio using a screen recording software with a timestamp. If this is done on a different computer, we will sync the audio by playing a succession of beeps at the beginning of the experiment and writing the time when that beep occured in a text file.

To be logged Each episode will have the following information logged in a folder named for the episode id:

- a png snapshot of the setting (what the wizard sees)
- a png snapshot of what the participant see (the setting with the target/landmark outlines)
- an ann.txt file which contains the (id,x,y) information for the target and landmark, if chosen
- a setting.xml file which contains the features of each segmented object
- a flagged.txt file is copied into the folder if the wizard explicitely flags the episode, or if the wizard clicks on the wrong objbect, in which case the x,y coordinates of the mouse click are written to the flagged.txt file
- a timestamps.txt file gives the start and end timestamps of the episode only if the episode finished by a mouse click (either correct or incorrect) by the wizard

We will do ASR after the experiment is complete.