An Image-To-Speech iPad App

Michael Maynord, Jitrapon Tiachunpun, Xiaojin Zhu,
Charles R. Dyer, Kwang-Sung Jun, Jake Rosin

University of Wisconsin Madison
Madison, WI, USA 53706
{maynord, tiachunpun}@wisc.edu
{jerryzhu, dyer, deltakam, rosin}@cs.wisc.edu

July 26, 2012

Abstract

We describe an iPad app which assists in language acquisition and development. Such an application can be used by clinicians for human developmental disabilities. A user drags images around on the screen. The app generates and speaks random (but sensible) phrases that matches the image interact. For example, if a user drags an image of a squirrel onto an image of a tree, the app may say "the squirrel ran up the tree." A key challenge is the automated creation of "sensible" English phrases, which we solve by using a large corpus and machine learning.

1 Background

The motivation for developing such an application is to assist those with language developmental disorders. There is room for improvement in current methods of language instruction, and with the emergence of new technologies comes the emergence of new options for language instruction. The iPad could evolve into a useful tool for language instruction because of it's ease of use, interactivity, and immediate response to input. The idea is that if a child is able to successfully engage with the material they are learning, they learn faster. To get a child to engage in language when they have little language skill, one could use interactions with images and present the corresponding phrases.

We developed this app in consultation with clinicians in the Waisman Center at the University of Wisconsin–Madison, who study language development and instructional methods for those with language developmental disorders.

Figure 1: A screen-shot of the app

2 Overview

A screen-shot of the app is given in Figure 1. From a user's perspective the application is kept simple. The user applies manipulations to images displayed on the screen of the iPad using their fingers, and the iPad reads back aloud to them an English rendition of the interactions of the images. For example, a user might drag an image of a monkey onto an image of a tree, and the iPad might read back aloud "monkey hangs from a tree," as shown in Figure 2.

Images can be place into contexts where each image is related under a given theme, see Figure 3. The advantage to having contexts is that the interactions of images within a context will both be more sensible, and be contextualized under the theme of the given context. The rationale being that contextualized interactions are more conductive to language acquisition. Some example contexts could be a kitchen context, a school context, a sports context, etc.

There are two main components to the project: user interface on the iPad, and backend phrase generation. Each component has constraints to work within, the reason for separating out the phrase generation from the iPad is that phrase generation would be greatly hindered by the limited capacity, both in storage and computational capacity, of the iPad.

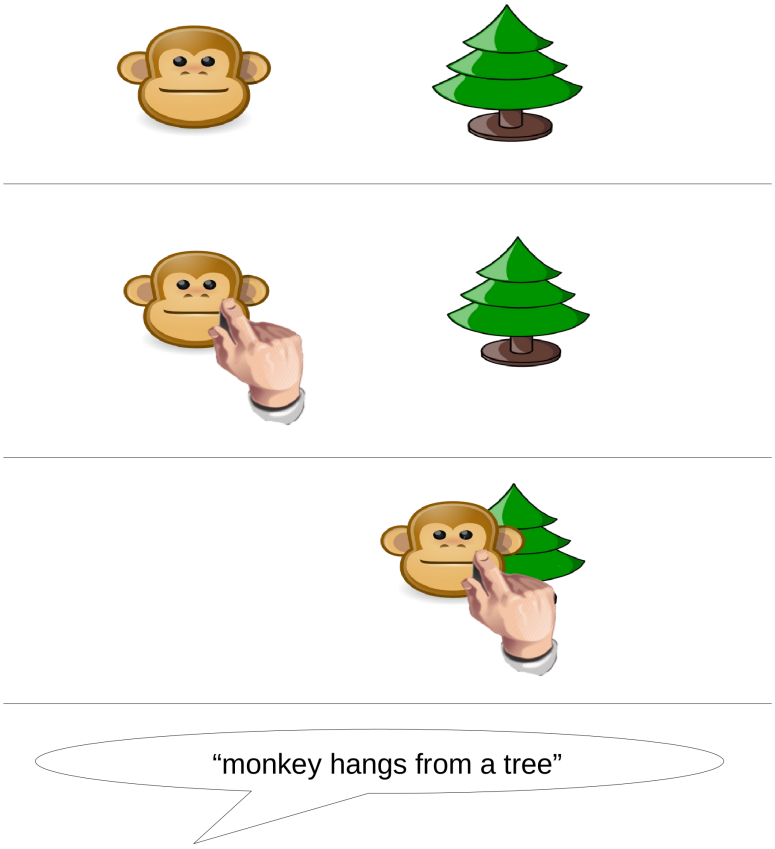

Figure 2: Example use

3 iPad User Interface

The user interface of the Image-to-Speech application presents the user with a scene containing various objects. These objects can be selected by the clinician beforehand.

The iPad presents a gesture recognition API which is used to determine when the image of one object has been dragged onto another. When this occurs a function is called and passed the names of the two objects as parameters. This function selects a phrase from the set of stored phrases matching the given nouns, displays it on screen, and invokes a text-to-speech engine to speak the phrase.

A callback function is also invoked when any image is first touched which invokes the text-to-speech call on the assigned name of the image.

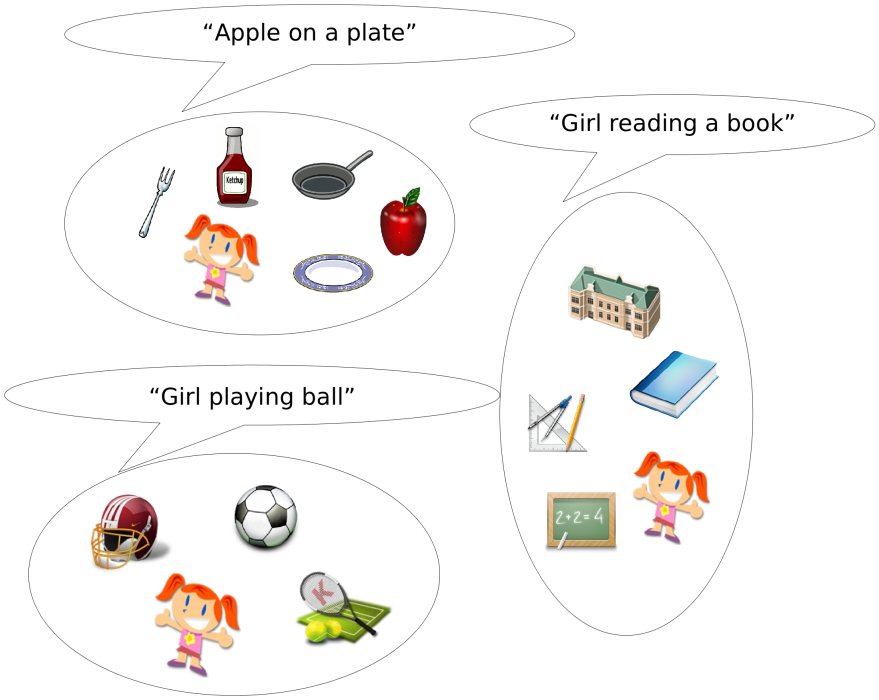

Figure 3: Example contexts: kitchen, school, sports

A task bar displayed at the top of the screen contains buttons to save or load a scene, to import individual images, or to remove them from the screen. Images from the iPad image library are presented for selection when the user chooses to import an image. A user can save an image from an external source to the iPad's image library and import it into the application. The task bar also contains a text field which contains the name of the currently selected image, this field can be altered to change the name of the image. An image which has been loaded onscreen but which has not yet been assigned a name simply has the name 'unnamed'. When a scene is saved it is the mapping between images and names that is stored.

On the iPad, each app has limited access to the file system. An app is given a directory of its own to work with – it is here that phrases are stored. All the phrases are stored in text files, each text file is named according to the phrases' subject noun and object noun (corresponding to the first and second images respectively, where the first was dragged onto the second). Within each file are up to 10 phrases involving the two nouns. The callback function chooses a phrase at random among the phrases in the file corresponding to the two nouns it was given. If the user produces an interaction onscreen involving a noun which

has been assigned a novel name, then there will be no phrases stored involving that noun, and the default phrase of "noun1 jumps over noun2" is returned.

4 Phrase Generation

Generating valid phrases from scratch is more involved than filtering preexisting phrases, so instead of generating phrases, they are filtered from an 87 GB corpus of 3, 4, and 5-grams collected by Google from text online. This corpus is meant to be a representative sample of all text online. An n-gram is an ordered list of n words, for example "monkey hangs from a tree" is a 5 gram. The Google corpus is used because it provides phrases of the desired length, which are sorted and indexed.

The process of filtering the corpus down to valid phrases occurs in multiple passes:

1. The first pass simply filters out invalid phrases. Phrases are filtered by several criterion:

   (a) First, they must start and end with nouns. Before filtering can begin, a list of all nouns to filter for must be compiled. This list should be as general as possible as in the first passes over the Google corpus nothing should be filtered out that might be useful at a later stage. For the first pass noun list, 77355 nouns were taken from the MRC psycholinguistic data-base.

   (b) Second, they must not contain any words listed in a blacklist. As the Google corpus is a representative sample of online text, much of which is obscene, each phrase must be filtered for obscenity. To do this multiple blacklists were constructed and merged together. Any phrase containing a blacklisted word is rejected. However, blacklists are not completely effective at filtering potentially obscene phrases. Before the final version of this application a more sophisticated means of profanity filtering should be employed, until then care should be taken so that no obscene phrases are presented to the end user.

   (c) Third, they must be grammatically correct. Grammatical correctness is determined first by labeling the parts of speech(POS) of each word in a given n-gram. The POS of a given word is the grammatical role it plays in the phrase. This can include roles such as multiple types of nouns, verbs, adverbs, adjectives, etc. Correctly labeling the POS of a given word in a given phrase is a sophisticated process, to do this we used the POS Tagger developed by the Stanford Natural Language Processing Group. After the words of a phrase have been assigned POS tags, the phrase is filtered according to a manually constructed regular expression over its POS tags. The regular expression accepts phrases of the form:

      noun verb adverb|IN DT noun

noun adverb|IN verb DT noun
noun “and” DT noun

2. The second pass reorganizes the file so that all n-grams of a given noun-pair are grouped together, rather than being split up by number of words, uppercase/lowercase, and sorted alphabetically.

3. Another pass produces phrases only for a small subset of nouns, and only up to 10 phrases per noun pair. It is the result of this third and last pass that is loaded onto the iPad.

Extracting phrases like this in passes makes implementation easier, simpler, and reduces turnaround time when some specification changes. For example, if one wants phrases for a different set of nouns to be loaded onto the iPad, one re-runs the third pass on this new noun list rather than starting from scratch with the Google corpus. To run all three passes from scratch starting with the Google corpus takes on the order of a week.

5 Future Work

This application originated with the question of how to improve language acquisition in those with language developmental disorders. One answer to that question was the use of technology to provide an interactive experience connecting visual interactions with language interactions. This application running on the iPad is such a technology, and it will be seen to what extent the current and future versions of this application are able to aid in language acquisition.

Acknowledgments

This work was supported in part by the National Science Foundation under Grant Nos. IIS-0711887 and IIS-1148012. In particular, MM and JT were undergraduate students participating in the NSF Research Experiences for Undergraduates (REU) program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to thank Jamie Murray-Branch and Julie Gamradt from the Waisman Center, University of Wisconsin - Madison, for their support and advice in developing this project.

References

[1] Xiaojin Zhu, Andrew Goldberg, Mohamed Eldawy, Charles Dyer, and Bradley Strock. A text-to-picture synthesis system for augmenting communication. In The Integrated Intelligence Track of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07), 2007.

[2] Andrew B. Goldberg, Xiaojin Zhu, Charles R. Dyer, Mohamed Eldawy, and Lijie Heng. Easy as ABC? Facilitating pictorial communication via semantically enhanced layout. In Twelfth Conference on Computational Natural Language Learning (CoNLL), 2008.
[3] Arthur Glenberg, Andrew B. Goldberg, and Xiaojin Zhu. Improving early reading comprehension using embodied CAI. Instructional Science, 2009.
[4] Andrew B. Goldberg, Jake Rosin, Xiaojin Zhu, and Charles R. Dyer. Toward Text-to-Picture Synthesis. In NIPS 2009 Symposium on Assistive Machine Learning for People with Disabilities, 2009.
[5] Arthur Glenberg, Jonathan Willford, Bryan Gibson, Andrew Goldberg, and Xiaojin Zhu. Improving reading to improve math. Scientific Studies in Reading, 2011.