COSC345 Software Engineering

Assignment 1 Report

Report by:

Winston Downes	(2872112)
Max Huang	(4427762)
Mitchie Maluschnig	(9692597)
Sam Paterson	(3175949)

16 April 2018

Abstract

Receptive Aphasia is a condition that is characterised by language comprehension deficits. The ‘tell-tale’ sign of people suffering from Receptive Aphasia is the production of speech which sounds ‘fluent’, i.e. it is rapid and laden with intonation and prose typical of regular speech; however, it is unintelligible. Our team aims to develop an app which will support patients suffering from Receptive Aphasia; the app will translate text to images (and vice versa) to facilitate communication between aphasic patients and non-affected persons. As images are devoid of the linguistic aspects with which patients have difficulty, they can be used to preserve and convey semantic meaning. The scope of our project will be restricted to iOS and watchOS, with support for iPhone 5 through to iPhone X and all Apple Watch models.

Introduction

Receptive Aphasia is caused by damage to Wernicke's area, which is located in the temporal lobe of the left hemisphere of the brain and is important in the comprehension of language. Thus, patients with this type of Aphasia have gross language comprehension deficits. While aphasics are able to utter phrases that sound very fluent, they nevertheless lack semantic meaning.

Our app aims to support patients suffering from Receptive Aphasia by aiding them in communicating with others around them. To this end, we intend to build an app that translates text to images and vice versa, to facilitate communication between aphasic patients and non-affected persons.

Our team consists of four members: Sam Paterson, Winston Downes, Mitchie Maluschnig, and Max Huang. In general, each team member will be contributing to the programming aspects of this project; however, the workload for this paper has been split up somewhat: Sam is mainly focusing on user interface elements; Winston is researching the particulars of Receptive Aphasia and working on sentence translation methods; Mitchie is working on image databases and investigating parts-of-speech recognition; and Max will be tackling report writing, general organisation of team elements, and app programming.

Our app will have inter-device compatibility between the iPhone and the Apple Watch. We will have support for all iPhone models spanning from the iPhone 5 to the iPhone X and all series of the Apple Watch.

After much discussion about risks of this project, our team is most concerned with the following: feature creep, the scope and size of the project, the security issues that could arise from cross-device communication, the size of the application on the device, the possibility of creating an app that is overly complicated and over-cluttered which can lead to a huge learning curve for users, the time constraints of the project, and problems we may run into whilst developing the app - mainly relating to the preservation of semantic meaning over the translation process, be it text-to-image or image-to-text.

Receptive Aphasia

Receptive Aphasia (previously known as Wernicke's or fluent aphasia) is a condition that can occur after damage is sustained to Wernicke's area of the brain, which is located in the posterior superior temporal gyrus of the left hemisphere of the brain, that typically results in gross language comprehension deficits (Knepper et al., 1989; see Figure 1). Although the leading cause of this condition is stroke (Knepper et al.) patients often do not suffer paralysis because the area of the brain implicated in Receptive Aphasia is not near the motor cortex of the frontal lobe, which is important in processing motor movement (c.f. Expressive Aphasia). Common symptoms include the inability to comprehend spoken speech as well as deficits in comprehending written text (Ardila, Bernal & Rosselli, 2016), and producing speech that is full of phonemic paraphasias (i.e. intended words are replaced with other words or non-words which sound similar; Binder, 2015). The speech of a patient often conveys little meaning and is therefore often unintelligible to other people; however, the speech is usually rapid, with intonation and prose that is typical of normal speech. This is the reason that this type of Aphasia was previously referred to 'Fluent' Aphasia. Although comprehension of spoken speech is traditionally recognised as the primary comprehension deficit, almost all patients will also exhibit deficits with written text as well. It is because of these two deficits that Receptive Aphasia is viewed as a disorder involving both the phonemic aspect of words (i.e. their sound elements) as well as at the lexical level (i.e. the 'form' of the word with respect to its sequence of parts; Ardila, Bernal & Rosselli).

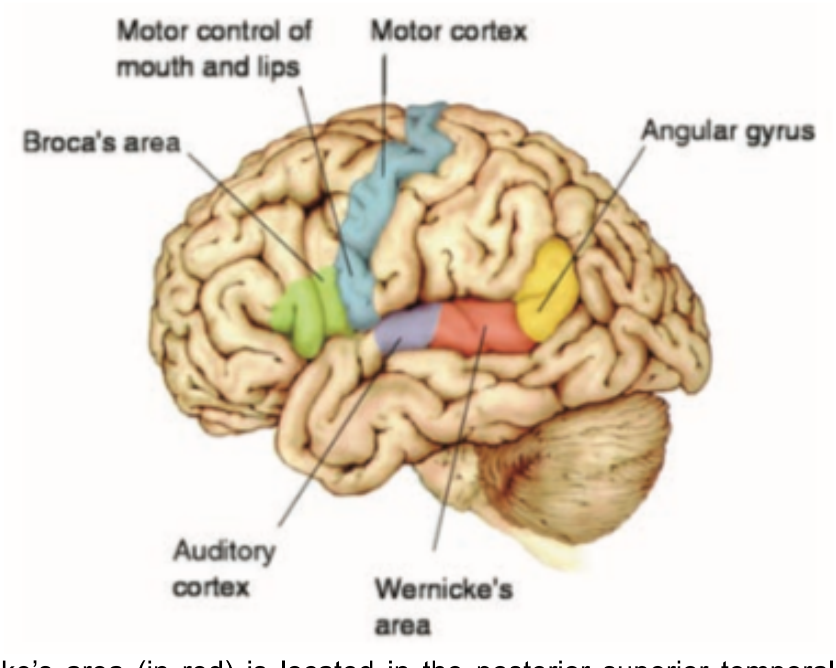

Figure 1. Wernicke's area (in red) is located in the posterior superior temporal gyrus of the left hemisphere.

App Support for Receptive Aphasia

The app we intend to build will target three symptoms of Receptive Aphasia: text comprehension, speech comprehension, and speech production. The app will take text that is input from a user who wishes to communicate with someone who suffers from Receptive Aphasia and translate it into a picture sentence. It is expected that the patient will be able to understand the intended meaning because pictures are devoid of the linguistic aspects, with which patients have difficulty (i.e. phonological and lexical elements). For example, if the user wrote “would you like to eat an apple?”, the app would translate that into a picture of a mouth baring its teeth as if to eat, with a picture of an apple to the right of it, followed by a question mark to the right of that - note that the typical left-to-right presentation of the items in the written sentence is preserved in the picture sentence. For speech comprehension, the app would dictate the spoken speech of the user and, again, translate that into a picture sentence. To communicate, the Aphasic patient could use these functions ‘backwards’, where a picture sentence is created by the patient by selecting the pictures of the nouns and verbs needed, and then that is translated into either text or sound for the non-affected user.

In an effort to elicit condition-focused, realistic feedback from people suffering from this disorder during our testing phases, we have been in contact with a member from the Community Care Trust (CCT), an Otago and Southland non-profit charitable trust that supports adults with intellectual disabilities. We hope to get in contact with an individual with Receptive Aphasia who would be interested to volunteer to help us test the app, thereby providing us with targeted feedback. We have also been in contact with a representative from the Disabled Persons Assembly (DPA), who have expressed an interest in this project and may potentially endorse our app.

Implementation

We will search for images from our own database that we will populate accordingly. In order to keep our text-to-image more accurate, each image will have one primary keyword as well as a part-of-speech identifier (i.e. whether the image describes a noun, verb, conjunction, etc.). If we decide to implement an ‘edit’ function for images generated from a sentence, each image in the database will also have an attribute that lists relative keywords to that particular image, either semantically or linguistically linked.

We intend to create our own database so that it will meet our set of specifications to allow for a more accurate search. If we were to use an existing image database, we may get several image results that we feel do not correctly represent the keyword. As the development of the app progresses, we expect the size of our database to increase accordingly - at some point, it will become too large to feasibly keep locally on the device, so we plan to store it online and have the device connect when a query is ready to be processed.

The Apple Developers Documentation includes an API for identifying parts of speech from a sentence. We will build on the NSLinguisticTagger class (Apple Inc., 2018) to split the input sentence into its parts-of-speech and then search our database using the appropriate keywords. This API is also able to identify the lemma (base word) of words in a sentence which we can then use to define tense.

App Design

When the user opens the app, they will be presented with a starting screen which gives them the option to use the text-to-image or the image-to-text translator, depending on the user. Once inside the text-to-image translator they will see a keyboard and a text box into which they will type the sentence they want to translate, and then press OK/Translate (see Figure 2). After the conversion has completed the user will have the option to review the result which includes changing the images if they so choose (to a more semantically accurate image) or changing the symbols after each image (changing the image would bring up another screen with suggested alternatives or a search functionality); this would be the completed translation screen shown to the Aphasic patient.

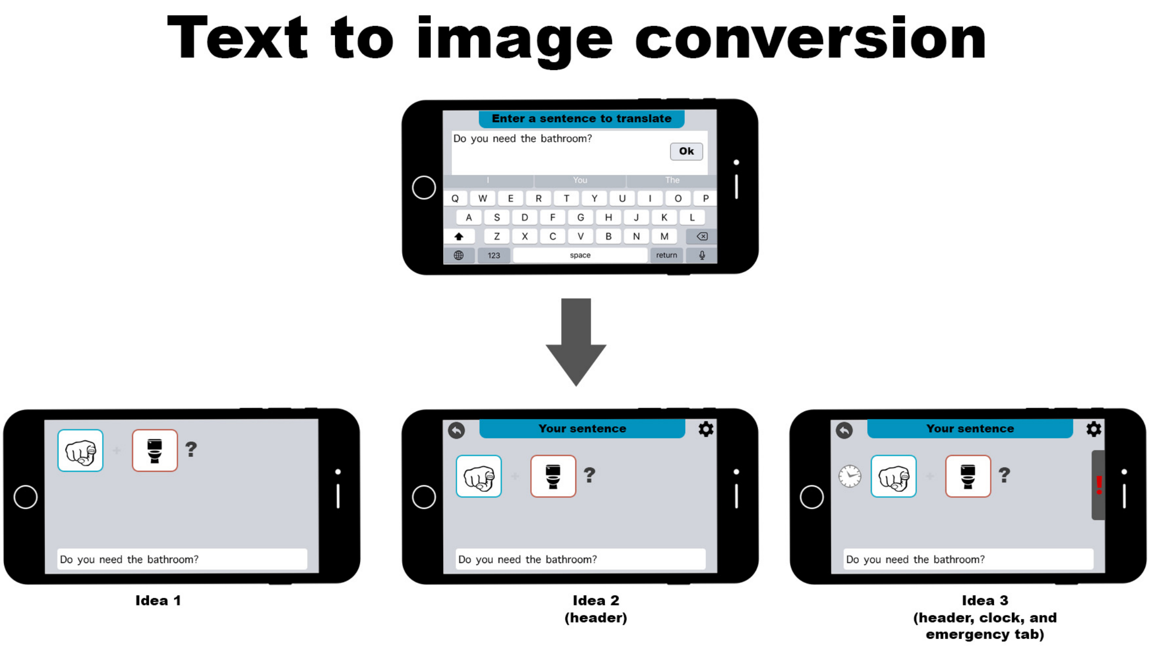

Figure 2. Text-to-image conversion showing various ways we could layout the UI.

A two-finger swipe will change from one mode to another (i.e. from text-to-image or image-to-text). Within the image-to-text translator, the user will be shown an array of images which they can choose (most frequently used will be shown first) this image selection will change and adapt (much like a predictive text function) allowing the user to create a sentence using multiple images with ease. The symbols after each chosen image can be changed by tapping on them (bringing up the available options). After tapping OK/Translate, the images will be translated into a text sentence and shown on the next screen, along with the images that were selected (down the bottom; see Figure 3).

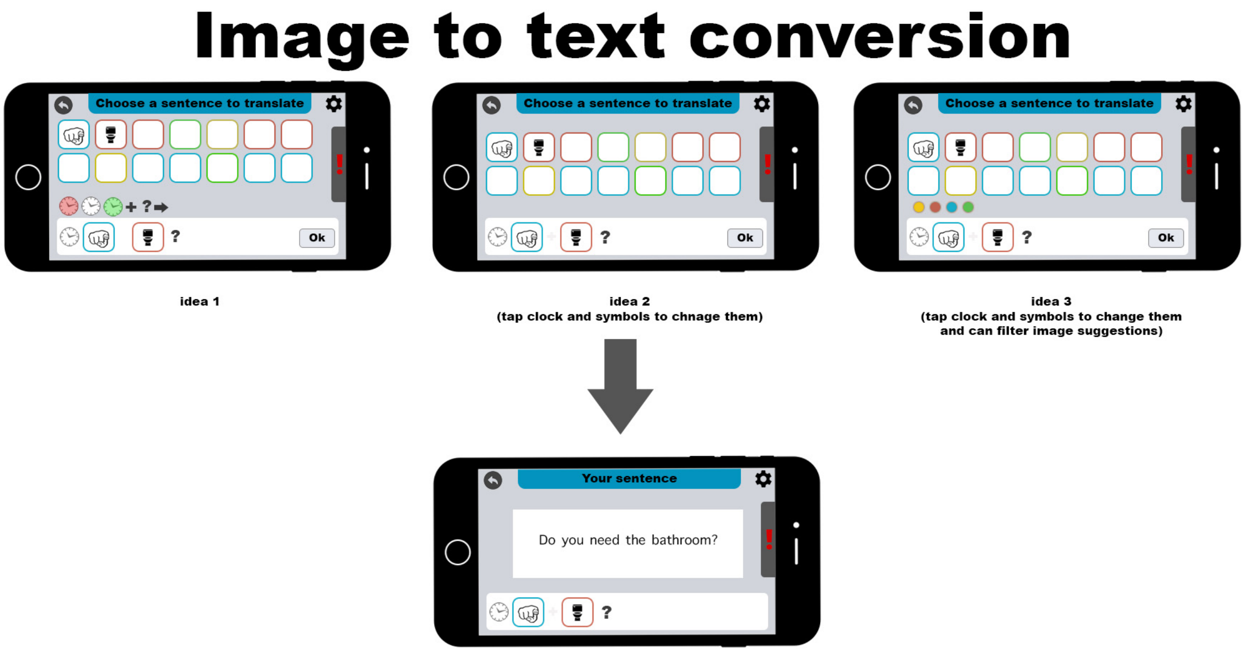

Figure 3. Image-to-text conversion showing how an image sentence could be constructed and possible different layouts.

There will be an emergency tab on the screen at all times which has buttons for different emergency options (e.g. connecting to emergency services or contact person) or even user determined options, such as “bathroom”.

We realised that the way a sentence is understood depends not only on context but also the tense. We decided that a good way to convey tense would be to use a colour coded stopwatch/clock; red for past, white for present, and green to indicate it is future tense. An image has the potential to mean an array of things depending on whether it’s a verb, noun, etc. so we are currently planning on having different coloured borders for the image which indicate the parts-of-speech. We will have options for symbols after each word which could be used to show the flow of sentence, conjunctions, or show questions

Scope – iPhone and Apple Watch

The scope of our project will be currently restricted to only iPhone and Apple Watch. The compatibility will span all models from iPhone 5 through to iPhone X (i.e. iPhone 5 / 5s / 6 / 6s / 7 / 8 / X) and all Apple Watch Series (i.e. Series 1 / 2 / 3). The main functionality of the app will remain on the iPhone whereas the Apple Watch will mostly be a supplementary device that is used for emergent situations (e.g. “call mum”, “call emergency services”, “need to use toilet”). We plan on keeping the Watch UI clean and simple, seeing as there is little screen real estate to work with.

Project Risk Analysis

We could encounter issues where we continuously come up with ideas and features but not enough time to implement them - we need to be careful and stick to the important aspects of this project, only introducing extra features if we have time.

Our project has the potential to become very large and complex. An ideal full-scale version of our app would allow non-affected persons to converse with an Aphasic patient with ease. The non-affected person would communicate by speaking into the device which translates the speech into text, and then into a string of images. Having read it, the patient can then construct a sentence out of images which are converted to text/speech.

Our team has set some (guideline) milestones for this project to keep us on track, making sure we have a functional app at key points along its development – specifically during the alpha and beta testing stages. We aim to have our initial prototype to work on one to two pre-determined sentences to showcase proof of concept, before moving on to increase the number of images available and creating a searchable image database.

To minimise chances of having an undeliverable project during key phases of the production pipeline we are going use version control and ensure that no team member ever works on the master branch and that at any point in time, there is always a working version of the app on the master branch. New features will be created and trialled on separate branches and easily scrapped if they aren’t promising.

When two devices are linked together the data to be sent between devices will either be a text string or a series of unique identifiers that relate to images in the database. This means we don’t have to send full-size images between devices and the data should be easy to encrypt.

Having thoroughly discussed it, the team has come to a consensus that it would not be feasible to keep all our images locally stored in the app (i.e. on the phone) as this would occupy quite a large portion of memory on the device. Instead, we have opted to investigate storing the majority of our images in an online database - this would allow us to optimise the storage on the local device. We plan to implement this by having a small database of frequently used images stored locally on the device and the rest stored online which can be accessed on a search-as-you-need basis, this also allows us to implement an ‘offline’ mode for the app - allowing the app to maintain functionality whilst not connected to the internet.

Image sentences in our app are made up of four components: the image itself, coloured borders which identify part-of-speech (nouns/verbs), a coloured clock symbol specifying tense, and symbols that identify conjunctions (‘+’ for ‘and’ or ‘|’ and fading/flashing images for ‘or’). In order to understand an image sentence, an Aphasic patient must be able to recognise these components and be able to use them when creating their own picture sentences. This will most likely create a steep-learning curve for the patient to overcome before being able to use the app. We will have a tutorial for the patient, which will teach them how to read and write using image sentences as well as other tutorial/instructions for non-affected persons.

A big problem we may run into is if a vague sentence is entered; for example, we know that, colloquially, “how are you” means “how are you feeling” but the verb (‘feeling’) in this sentence is implied in our day-to-day language. We will have to overcome this by either designing our algorithm to identify these special cases or prompt the user to be more specific if the algorithm is not able to convert the sentence.

References

Apple Inc. (2018). NSLinguisticTagger. Retrieved April 10, 2018, from https://developer.apple.com/documentation/foundation/nslinguistictagger

Ardila,A., Bernal, B. & Rosselli, M. (2016). The Role of Wernicke's Area in Language Comprehension. Psychology & Neuroscience, 9, 340-343.

Binder, J.R. (2015). The Wernicke area: Modern Evidence and a Reinterpretation. Neurology, 85, 2170-2175.

Knepper, L. E., Biller, J., Tranel, D., Adams, H. P. Jr., & Marsh, E. E. III. (1989). Etiology of Stroke in Patients with Wernicke's Aphasia. Stroke, 20, 1730-1732.