Modules used

# Pyttsx : Python text to speech converter

This module requires an installation of a speech engine driver. In our application we used espeak driver for linux. Any application which needs to use pyttsx invokes the pyttsx,init() factory function. During construction of the engine object, the engine intilaizes the driver object responsible for loading a speech engine driver implementation. After the engine object is constructed we can use that object to perform various tasks like text to speech conversion, adjust the voice output, change the gender of the render and so on.

The init function actually creates an object of pyttsx.engine.Engine class which has the following important methods we used

1. **setProperty() : Sets the rate of speech and volume**
2. **say(*text : unicode*, *name : string*) : To speak out text**

Another important class we can initialize from the module is pyttsx.voice.Voice. An object of this class has methods to perform the following tasks

1. To set the language of speech
2. To set the gender and age of the renderer

# Pytesseract : Tesseract OCR for PYTHON

Python-tesseract is an optical character recognition (OCR) tool for python.  
That is, it will recognize and "read" the text embedded in images. Python-tesseract is a wrapper for google's Tesseract-OCR ( http://code.google.com/p/tesseract-ocr/ ). It is also useful as a  
stand-alone invocation script to tesseract, as it can read all image types supported by the Python Imaging Library, including jpeg, png, gif, bmp, tiff, and others, wheras tesseract-ocr by default only supports tiff and bmp. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.

In our project, we perform sharpening of the image using OpenCV before passing it to tesseract. Hence we can improve the confidence of the output. Even though this module is claimed to be a wrapper for tesseract-OCR majority of the methods of tesseract-OCR are not available in Pytesseract. There is another package called Pytesser which does the same functionality.

The main function we used from this module is

# OpenCV : Image Processing IN PYTHON

OpenCV libraries comes in handy when we do anything in image processing. We describe some of the major tasks we performed by using methods from this module.

We used cv2.imread() and cv2.imshow() to read and display images.

cv2.cvtColor(crop\_img, cv2.COLOR\_BGR2GRAY) – To convert RGB image to grayscale to process

Sonal – Fill in here

# PyDictionary : Python Dictionary Module

This module and it’s methods helps us in getting the meanings, translations antonyms and synonyms of a word.

The usage is as follows

Create an instance of dictionary by

dictionary= PyDictionary()

Now we can use this dictionary object to get meaning and translation by using the meaning method and translate method on the object.

# Urllib, Simplejson

**JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.  JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. When we need to interact with google, Wikipedia to search images and data in our project we use the standard API’s provided by the respective company. We open the url by using functions from the urllib library of python. These API’s are a software-to-software interface, not a user interface. With APIs, applications talk to each other without any user knowledge or intervention. When we want to interact with an API in Python (like accessing web services), we get the responses in a form called JSON.

To interact with JSON, we can use the json and simplejson modules in python

Once JSON object is loaded into python by using the above modules, it just becomes like a dictionary.

JSON data structures map directly to Python data types, which makes this a

powerful tool for directly accessing data without having to write any XML

parsing code.

For example

data = json.load(urllib2.urlopen(url))

gets the entire dictionary of the url to data. From this dictionary in python data, by applying the keys specified by vendor say google, we can extract the required information.