Smartext Report

Lalagkos Lysandros, Basho Alexander, Michelioudakis Theodoros

# Product Description

Smartext is a notepad application that offers accessibility and Machine Learning features designed to facilitate faster and more comfortable note taking for everyday tasks. Its features include classic note taking features such as copy-paste, cut, find and replace text, new, rename, save, save as and load files. Novel features include encrypting and decrypting saved files, speech to text writing and text to speech audio, Optical Character Recognition and Handwriting Recognition from digital images, Sentiment analysis, searching for WikipediaTM lemmas, finding synonyms or antonyms for chosen words and last but not least finding definitions for chosen words. It also includes two visual themes, Light mode and Dark mode. Smartext is written in the Python programming language, using JetBrainsTM PyCharm with the Anaconda Python 3.7.4 interpreter. Internally, It follows an Object Oriented Programming structure and was coded using a hybrid method combining Waterfall Development, SCRUM and Extreme Programming. It is intended to be used exclusively with Microsoft’sTM Desktop Windows 10 Operating System

# Features

## Encryption/Decryption

Along with the usual file operations, Smartext can also encrypt .txt files using the AES specification, with a password set by the user, and can also decrypt .aes files with the according password. This feature offers unparalleled security for the user’s important files, while also not limiting the portability of said files, as they are not hidden, but rather encrypted.

## Speech to Text

Using Google’s powerful API, Smartext can use a device’s microphone to listen to the user and transcribe the spoken language to text. This allows for greater accessibility to note taking, especially for the visually impaired, and offers more options with which the user can interact with the application. It requires an active Internet connection.

## Text to Speech

Using a Python library, Smartext can dictate all the written text to the user. This is another accessibility feature that is intended to be used as a proof writing tool, by the visually impaired or to allow for another form of “reading” some text. It does not require an Internet connection.

## Optical Character Recognition

Using a Python library, Smartext can recognize typed or printed text from an image to output digital text, allowing for convenient conversion of images to text

## Hand Writing Recognition

Using pre-trained Machine Learning models, Smartext can recognize hand written text, from an image and convert it to digital text. The limitation of this feature is that it can only do so line-by-line , meaning that the images must contain only one line of hand written text. This allows for easy conversion of hand written text to digital text without the use of external tools or applications.

## Wikipedia Lemmas

Using a Web-Scraping Python library along with the Wikipedia API, Smartext can search for key-words (lemmas) and returns wiki articles to the user. This enhances the note taking experience as it allows for convenient lookup of terms and information, without the need of an browser. It does however require an active Internet connection

## Synonyms/Antonyms/Definitions

Using a Python Library, Smartext can search for synonyms, antonyms and definitions of chosen words, so that the user can access that information without needing a dictionary or browser.

## Sentiment Analysis

Using pre-trained Machine Learning algorithms, Smartext can suggest a prediction of the impact some text will have to the reader. This can help the user clear up ambiguous text to have a definite positive or negative tone. The limitations of this feature is that slang, cultural references and variations, sarcasm and other contextual content of some text are not recognised and can therefore confuse the model.

## Dark and Light visual themes

Smartext can be set to have either a Light or Dark mode. Light is mostly white and bright, accommodating bright environments, and Dark uses Black and Yellow for use in dimly lit environments.

# Installation

Smartext comes packaged in a .zip file. To use it, the user must decompress the file, open it and run Smartext.exe. The overall directory requires some 4.4GB of hard drive space, due to the use of NLTK’s data corpus, that is over 3GB, and the use of machine learning models, that also require a lot of storage space.

# User Manual

## File operations

### New File

Creates a new, empty text file.

### Save As

Save the current file the user is working on with a chosen name , in a chosen directory.

The default directory is Smartext/texts, however the user can choose any directory.

### Save

Saves the current file the user is working on.

### Encrypt

The user chooses a saved text file. The application then asks for an encryption password (key) and applies AES encryption to the file. It outputs a .aes file with the same name as the original .txt file

### Decrypt

The user chooses a .aes file. The application then asks for the password that the file was originally was encrypted with (key). If the password is correct the file is decrypted to a .txt file. If it is incorrect the decryption fails and there is no output.

### Rename

The user can choose a text file that they want renamed. Then they are asked for the new name.

## Edit Operations

### Undo

Reverts the latest changes / additions the user has made to the text.

### Redo

Reverts the latest Undo operation the user has made.

### Cut

Cuts some chosen (highlighted) text, the text remains in memory and can be Pasted elsewhere.

### Copy

Copy some chosen (highlighted) text, the text remains in memory and can be Pasted elsewhere.

### Paste

Writes the most recent in-memory text.

### Delete

Erases the chosen (highlighted) text.

### Select All

Highlights the entirety of the text the user is working on.

### Clear All

Deletes the entirety of the text the user is working on.

## Dictionary

### Definition

Creates a new window where the user is asked for a word they need the definition for. It then searches and returns the most relevant definition.

### Synonyms

Creates a new window where the user is asked for a word they need synonyms for. It then searches and returns the most relevant synonyms.

### Antonyms

Creates a new window where the user is asked for a word they need antonyms for. It then searches and returns the most relevant antonyms.

## Vision modes

### Light mode

Sets the colour scheme of the application to mostly whites and light grey.

### Dark mode

Sets the colour scheme of the application to mostly blacks and darker yellow.

## Text Alignment

### Left

All text is bound to the left border of the document

### Right

All text is bound to the right border of the document

### Centre

All text is centred horizontally between the left and right border

### Justify

Text is bound to both borders, so that it completely fills all lines except the last.

## Wikipedia

Creates a new window that asks for some lemma to lookup on Wikipedia. Returns the most relevant article. If the lemma is ambiguous, it returns the disambiguation (all the relevant lemmas).

## Sentiment Analysis

Creates a new window with a “Enter new text” button. Upon pressing the button a new window with a text box appears. The user inputs any text and the application will analyse the text and return a “Positive” or “Negative” prediction on the original window with the button.

## Speech to text

Upon pressing the application activates the device’s microphone and waits for the user to say something. When the user is done talking the application writes the text it detected.

## Text to speech

Upon pressing the application will start reading all the text in the text area. It dictates said text through the device’s default audio device.

## Optical Character Recognition

Upon pressing the application asks the user to choose an image. Then the image is analysed to detect and typed/printed (NOT handwritten) text on the image and returns it to the user.

## Hand Writing Recognition

Images containing lines of handwritten text should be placed in Smartext/HTR/WSdata/. Upon pressing the HTR button the user should select a file named after the image they would like to analyse, and a new window will appear with the words the program was able to detect.

# Documentation

## Smartext.py

The main file of the program, it is used to initiate the main window and holds the basic textpad functionality and GUI. All peripheral files are called from here, following OOP guidelines. The entirety of the GUI is created using Tkinter, a Python module that allows for very customizable interfaces. It is the skeleton of all the code.

## AEScrypto.py

This file contains simple functions used to encrypt and decrypt .txt files. It is based on a Python module PyAesCrypt which handles the raw encryption and decryption operations. When encrypting or decrypting, the GUI for the file dialog and password are handled by Smartext.py

## wikiCrawler.py

The file contains a simple function that is used to return Wikipedia articles or disambiguations. It is based on the official Wikipedia API in a practical Python module. The GUI is handled by Smartext.py

## syn\_anto.py

The file contains functions that return synonyms, antonyms or definitions of words. It is based on a massive word corpus contained in the NLTK Python module, a multifunctional tool for Natural Language Processing. The GUI is handled by Smartext.py

## sentiment\_analysis.py

The file contains multiple functions that handle Sentiment Analysis operations, and returns the prediction. It is based on NLTK and Pickle Python modules, both of which are for Natural Language Processing. The GUI is handled internally, as it is a rather large file.

## OpticalCharacterRecognition.py

The file contains a large function that returns text from an image. It is based on the OpenCV and Tesseract Python modules, the first for reading the image and the second for extracting the text. The GUI is handled internally, as a popup is required.

## NEWt2s.py

The file contains a simple function that dictates the text written in the text area. It is based on on the pyttsx3 Python module, a text-to-speech library. This feature does not require additional GUI elements

## speech2text.py

The file contains a simple function that will listen to the device’s microphone and transcribes the spoken words to text. It is based on the SpeechRecognition Python module, a library for speech-to-text library that supports multiple API’s from companies such as Google and Microsoft. Our implementation utilizes Google’s service, and therefore requires an active Internet connection to function. This feature does no require additional GUI elements.

## wikiCrawler.py

The file contains a simple function that when provided with a lemma either returns a summary of the article of said lemma or a disambiguation page. The GUI is handled by Smartext.py.

## HTR(Hand writing recognition)

HTR could stand an application all of its own, as it requires a main file and multiple peripheral files to function. It can be set up to run separately in a console, or with it own GUI. It essentially is two distinct programs. The first is a line segmentation program that extracts separate words from a line, and the other is a true HTR program that can “read” Handwritten words. The two combined make this module, which takes lines, segments them, and the iterates through each word to return a digitized text of a handwritten line.