# DIGITAL ASSISTANT BASED ON DESKTOP APPLICATION USING SPEECH RECOGNITION AND TEXT TO SPEECH WITH TYPOCORRECTING

Alvon Danilo Sukardi
*Computer Science Department*
*(School of Computer Science)*
*Bina Nusantara University*
Jakarta, Indonesia 11480
alvon.sukardi@binus.ac.id

Gerry William Nanlohy
*Computer Science Department*
*(School of Computer Science)*
*Bina Nusantara University*
Jakarta, Indonesia 11480
gerry.nanlohy@binus.ac.id

Michelle Christian Bell
*Computer Science Department*
*(School of Computer Science)*
*Bina Nusantara University*
Jakarta, Indonesia 11480
michelle.bell@binus.ac.id

*Abstract* — **Digital Assistant is actually a familiar thing that we often encounter in the world of informatics and information technology. The principle is the same when you listen to the other person, but the other person is an electronic device in the form of a digital assistant. However, all that technology is most likely only accessible to normal people or at least those who can control technology with their physical fitness. What about the disabled and the elderly? Can they enjoy the advancement and sophistication of technology, and get help from it? Of course, not all of these technologies can be controlled and can help their daily lives. Therefore, in this paper, the author makes a desktop application-based digital assistant product called Alpha, which can be run through a computer OS (Operating System). Alpha Digital Assistant can be run using either voice commands or typing. For voice commands, users can give orders to assistants and can communicate with assistants and interact with each other through questions and answers. In addition, digital assistants can also be run with commands via computer typed input. No need to worry, the application is equipped with a typo correction feature that can correct typos in text.**

***Keywords — digital assistant, desktop application, speech recognition, text-to-speech, typocorrecting***

## I. INTRODUCTION

Technology is a science that studies skills in creating tools or devices to processing methods to help complete various human jobs. Along with the development of technology, nowadays many types of smartphones have been found, ranging from those with a 'querty' keyboard to touch screens with various operating systems such as iOS and Android, each operating system continues to be developed from time to time[1].

The development of smartphone technology as it is now, apparently cannot be felt by people with disabilities such as the physically disabled and blind. This is caused by the difficulty of operating a smartphone which is mostly implemented with a touch screen and as we know, people with disabilities can only use certain senses in their daily lives. Therefore, to assist persons with disabilities in operating touch screen smartphones, we created a desktop-based digital assistant application using speech and text recognition with typo correcting features designed in Python and the Tkinter library.

Digital Assistant is a software agent which is also an intelligent agent that performs a task with some specific directions from the user through command instructions either by voice (speech recognition) or typing (text-to-speech). This software evolved from the concept of a virtual personal assistant, a cognitive assistant that can learn and organize. Digital Assistant combines approaches in traditional artificial intelligence to try to create a personal assistant program that can evolve through interaction with the user.

This software uses intelligent techniques and behavioral learning to show users that even a computer can be smart. The purpose of developing the Digital Assistant program is to create an extraordinary experience which can help people to be more efficient in their daily lives. This is a major breakthrough to help solve problems experienced by people with disabilities, for example blind people who really want to find out about an information but cannot see, then the assistant role can be used to find the desired information through speech recognition[2].

But to make a program in the form of a Digital Assistant run smoothly is not an easy thing. There are many challenges that the author faces. These challenges include: the program cannot run properly which is influenced by the specifications of the hardware used, adjustments to the pronunciation of every word spoken in voice command instructions (speech recognition), typo correcting features on typing command instructions (text recognition), and display for an attractive and efficient user.

The rest of discussions carried out in this paper will include: Chapter 2 will discuss research on digital assistant that has been conducted, Chapter 3 will discuss the methods and algorithms used by the author in this paper, and Chapter 4 will contain a discussion of the methods and algorithms used in this paper and Chapter 5 will include the findings and conclusions from this paper.

## II. RELATED WORKS

1. A Conversational Digital Assistant for Intelligent Process Automation (Yara Rizk, Vatche Isahagian, Scott Boag, Yasaman Khazaeni, Merve Unuvar, Vinod Muthusamy and Rania Khalaf, 2020)

This lesson discusses the automation of robotic processes through digital assistants as a leading approach to automating tasks. Completing daily life work using this approach reduces costs in software updates, wide accessibility, and interactive conversations.

2. Digital Assistant for The Blind (Prince Bose, Apurva Malphtak, Utkarsh Bansal, Ashish Harsola, 2017)

   This lesson discusses technological advances that make it possible for the blind to use the internet or all forms of digital information in the form of hardware and software, apart from relying on Braille and keyboards which are expensive and rare. The author also designed a voice control system for the visually impaired that transmits information in audio form. This allows users to receive and send emails, access daily news, weather forecasts, set reminders and alarms, and take notes.

3. Digital Assistant PC Media, Vol 2, Hal.20 (Verydias Aditya, 2015)

   This lesson explains the comparison of the features found in the current digital assistant, starting from Google Assistant, Alexa, Cortana, Siri and so on.

## III. Tools, Framework, And Method

### A. *Python*

Python is one of the most popular programming languages and is widely used by programmers today. The popularity of this programming language is not without a reason. Python is known as a programming language that is very close to human language (high-level programming language), making it easier for developers and even ordinary people to understand the code syntax. Furthermore, Python is well-known as one of the languages that are widely used in artificial intelligence applications, one of which is NLP (Natural Language Processing).

The author uses this programming language because there are many libraries related to Natural Language Processing technology, making it easier to build Digital Assistant applications in desktop applications.

1. Pyttsx3

   Pyttsx3 is a library of the python programming language for the purposes of text-to-speech recognition, in which the computer will convert the input text from the user into a speech from the AI (Artificial Intelligence) itself. First the program will initialize and construct the sound engine from this library. Then, the application can output and stop the sound from the previously constructed sound engine[3]. Apart from outputting and stopping the sound, there are various functions that this library can perform, including saving the sound into a file, changing the engine sound, changing the volume, and many more[4].

   The author uses this library to assist users in communicating with the Digital Assistant. The program will convert the string stored by a variable into a voice that is spoken to the user so that there is direct communication between the user and the Digital Assistant.

2. WebBrowser

   The WebBrowser module provides the user with a view-to-view web-based content. In most cases, simply calling the open() function of this module is sufficient to open it[5].

   The author uses this library so that users can open the web or certain web content after giving orders, either through voice media or typing to the Digital Assistant.

3. Random

   This module is useful for generating random numbers or results for a particular function or variable using the distribution function[6]. In the program, the author uses this library to get random results from a choice of words through a list of strings from python with the choice() function. The function will return an index error if the list is empty[6].

4. SpeechRecognition

   Speech Recognition requires 3 components for analysis purposes, namely preprocessing, speech features, and postprocessing[7]. As the name suggests, this library is used to perform voice detection by providing an API (Application Programming Interface) and engines from several well-known companies, such as Google, IBM (International Business Machines Corporation), and others[8]. This library is one of the most popular speech recognition libraries and is widely used by developers. When compared to SpeechPy, this library is much better[9]. So that users can use the microphone to communicate with the Digital Assistant, this library requires the Pyaudio library. Pyaudio is a library that can be used across multiple platforms to record and play specific audio[10].

   The author uses the SpeechRecognition library so that users can communicate with the Digital Assistant via voice. This is supported by an increase in voice accuracy through the functions that have been provided, so that users can still talk even in crowded conditions.

5. WolframAlpha

Speech Recognition requires 3 components for analysis purposes, namely preprocessing, speech features, and postprocessing[7]. As the name suggests, this library is used to perform voice detection by providing an API (Application Programming Interface) and engines from several well-known companies, such as Google, IBM (International Business Machines Corporation), and others[8]. This library is one of the most popular speech recognition libraries and is widely used by developers. When compared to SpeechPy, this library is much better[9].

So that users can use the microphone to communicate with the Digital Assistant, this library requires the Pyaudio library. Pyaudio is a library that can be used across multiple platforms to record and play specific audio[10].

The author uses the SpeechRecognition library so that users can communicate with the Digital Assistant via voice. This is supported by an increase in voice accuracy through the functions that have been provided, so that users can still talk even in crowded conditions.

6. NLTK

NLTK is one of the popular multi-platform libraries of the Python language that allows developers to build programs using human language data. This library provides functions to perform text processing, such as classification, tokenization, stemming, and many more[11]. The author uses this library to create a special.

Digital Assistant corpus so that users can still perform certain commands without worrying about typing typos. The technique used to overcome typos entered by the user is to use the Jaccard Distance formula and n-grams or more precisely unigrams, as well as tokenization to divide sentences into lists of strings where the value is each word inputted by the user. These three techniques use the functions provided by the nltk library, namely jaccard_distance, ngrams, and word_tokenize.

The n-gram model is a sequence of sequential symbols that can be characters, words, or other continuous symbols. For example, 1-gram (or unigram) is one symbol, 2-symbol (bigram) is a sequence of two symbols, and 3-gram (trigram) is a sequence of three symbols, and so on [195264798]. This study uses the previous 1 character to predict the next ($tn|tn{-}1$). Processing is then performed as tokenization into individual tokens by removing punctuation and splitting each word into a string at each list index.

Jaccard distance is a formula to measure the dissimilarity between two values with the formula:

$$Dj(A,B) = 1 - J(A,B) = (|AB| - |AB|)/|AB|$$

By combining this formula with a unigram, the program can measure the dissimilarity between 2 strings, namely the wrong word and the right word. Of all the existing values, the program will take the shortest distance or the smallest value between the strings and correct the wrong words.

7. Wikipedia

Wikipedia is a Python library that is directly linked to Wikipedia and serves to access and parse data such as article conclusions, Wikipedia searches, etc[12].

The author uses this library to answer certain questions from users, which WolframAlpha cannot answer.

8. Datetime

Datetime is a Python library for accessing the current date and time[13]. The author uses this library so that users can ask questions about the date and time at that time.

9. Sys

Sys is a library for accessing and maintaining the interpreter of Python itself[14]. The author uses this library so that the desktop application can be closed if the user wants it.

10. Os

Os is a Python library to access our system by using the provided functions[15]. The author uses this library so that users can shut down their PC (Personal Computer) or laptop by simply ordering the Digital Assistant.

11. PyWhatkit

PyWhatkit is a Python library that has many features related to web browsers. Some of these include sending messages to WhatsApp groups or contacts, converting images into ASCII (American Standard Code for Information Interchange) art, converting strings into handwriting, and much more[16].

The author uses this library so that users can access certain songs or videos through the Youtube platform.

12. Tkinter

Tkinter is a Python library used to build a GUI (Graphical User Interface) for desktop applications. This library can set the interface layer, the size of the application geometry, manage the components that exist in a desktop application, and others[17].

The library itself has weak documentation, but materials such as references, books, and tutorials are still comprehensive. In addition, this library is also fast in rendering applications[17].

The author uses the library to build a GUI display for the Digital Assistant application. Although, this library is famous for having an old-fashioned look[17], the author managed to make the Digital Assistant application more attractive, modern, and user-friendly. This is because the author uses self-made image assets and inserts them into the application as needed.

13. PyInstaller

PyInstaller is a Python library for creating a package or folder that contains all the requirements needed to create a python application, such as the python library and interpreter[18]. By using this library, the user does not need to install the Python library in the application.

The author uses this library to convert a .py file (python file) into a single .exe file (application file), so that other users can access the application much more easily. The author also changed the logo of the icon and removed the console from Python to make it simpler and more attractive.

### *B. Visual Studio Code*

Visual Studio Code is a code editor that supports a variety of programming languages, including HTML (HyperText Markup Language), CSS (Cascading Style Sheet), and JavaScript. In addition, the tool is compatible with all PC operating systems, including Windows, macOS, and Linux. To use Python in Visual Studio Code, users must first download the Python extension, which is one of the most popular extensions in the program [https://code.visualstudio.com/docs]. Before starting to code the program itself, users must first create their own folder or workspace in Visual Studio Code[19].

The authors of this study used this tool to accommodate projects and build disease prediction programs in combination with other methods. When using Visual Studio Code to build projects, authors gain many advantages, including the ability to connect directly to GitHub, countless extensions that make it easier for writers to write programs, a cleaner project workspace, and more.

### *C. NSIS (Nullsoft Scriptable Install System)*

Most desktop applications today come from installers. After the installer is installed to the user's laptop or PC path, the user can immediately run the application (.exe) file directly without being confused about having to install other requirements, because all requirements have been inputted in the [NSS] installer. NSIS (Nullsoft Scriptable Install System) allows developers to create a small win32 installer for free[20].

The author uses this tool because the author wants to make an installer of the Digital Assistant application, so that users do not need to install the requirements needed to run the application. Users only need to install the installer to the desired path explorer because everything is already available in one bundle packages.

## IV. Result And Discussion

### *A. Design Overview*

At the design stage of the Alpha Digital Assistant application prototype, the thing to do is to apply any requirements that have been previously determined during the research stage. The requirements for the Alpha Digital Assistant application are:

1. The Alpha Digital Assistant application is deployed or based on a desktop application.

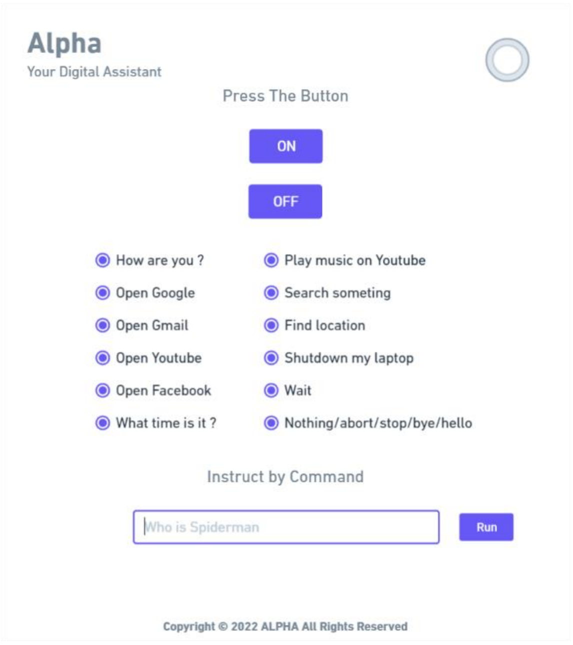
2. The Alpha Digital Assistant application has a text-to-speech feature, so users can carry out command instructions via typed input. So in the design there must be a layer for typing text.

3. The Alpha Digital Assistant application has a speech recognition feature so that users can perform command instructions via voice input. So in the design there must be a layer to activate (on button) and disable (off button) this feature.

4. The Alpha Digital Assistant application has a minimalist, attractive and efficient theme, but still prioritizes its main functions. So the color and shape design of the character must follow a minimalist, attractive and efficient theme but the main function must continue to run.

The prototype of the Alpha Digital Assistant application was made using Figma. Figma is a web-based vector graphics editor and prototyping tool, with additional offline features enabled by the desktop application for macOS and Windows. As for the mobile application, Figma is also available on Android and iOS. Figma on both platforms allows editors to view and interact with team worksheets to design prototypes in real-time.

The author designed the appearance of the Alpha Digital Assistant application by creating a minimalist, attractive and efficient theme, but still prioritizing its main function by taking advantage of the advantages of Figma's features, so that work becomes much easier and more flexible so that the Alpha Digital Assistant application gets a good response. from the user. In addition, the author has redesigned the appearance of the Alpha Digital Assistant application 3 times to get the best display, not only that the author also designed a logo in the form of a robot image. Attached is the application design along with the Alpha Digital Assistant logo from figure 1, figure 2, figure 3 and figure 4:



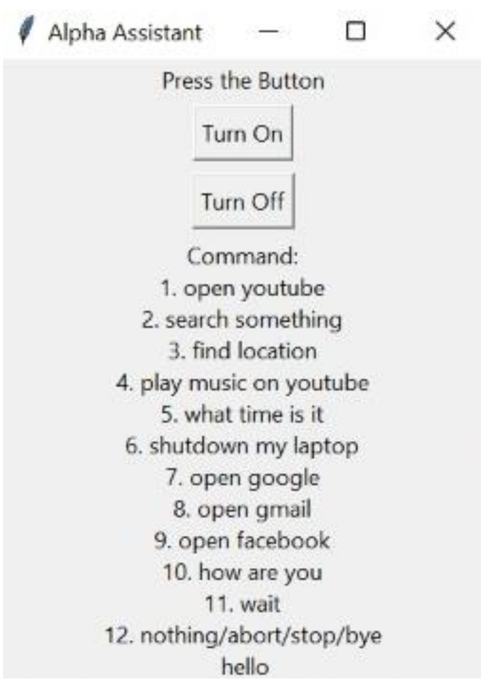
Alpha
Your Digital Assistant
Press The Button
ON
OFF

| | |
|---|---|
| ● How are you ? | ● Play music on Youtube |
| ● Open Google | ● Search someting |
| ● Open Gmail | ● Find location |
| ● Open Youtube | ● Shutdown my laptop |
| ● Open Facebook | ● Wait |
| ● What time is it ? | ● Nothing/abort/stop/bye/hello |

Instruct by Command
Who is Spiderman
Run

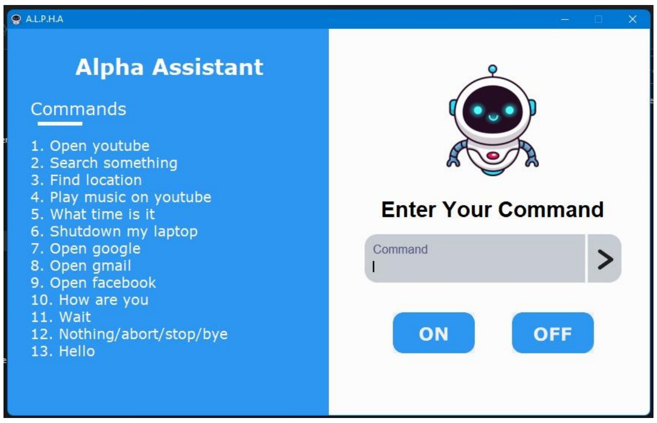
Copyright © 2022 ALPHA All Rights Reserved

**Fig 1.** First Design of Alpha Assistant Digital Prototype



Alpha Assistant
Press the Button
Turn On
Turn Off
Command:
1. open youtube
2. search something
3. find location
4. play music on youtube
5. what time is it
6. shutdown my laptop
7. open google
8. open gmail
9. open facebook
10. how are you
11. wait
12. nothing/abort/stop/bye
hello

**Fig 2.** Alpha Assistant Digital Second Design



A.L.P.H.A
Alpha Assistant
Commands
1. Open youtube
2. Search something
3. Find location
4. Play music on youtube
5. What time is it
6. Shutdown my laptop
7. Open google
8. Open gmail
9. Open facebook
10. How are you
11. Wait
12. Nothing/abort/stop/bye
13. Hello

Enter Your Command
Command
ON
OFF

**Fig 3.** Alpha Assistant Digital Third Design

**Fig 4.** Alpha Assistant Digital Application Logo (Own Asset)

Then for the code for the Alpha Digital Assistant application program, the author made it using Visual Studio Code and the Python programming language. The author uses Visual Studio Code to accommodate projects and build programs in combination with other methods. When using Visual Studio Code to build projects, authors gain many advantages, including the ability to connect directly to GitHub, countless extensions that make it easier for writers to write programs, a cleaner project workspace, and more. The author uses the Python programming language because there are many libraries related to Natural Language Processing technology, making it easier to develop desktop application-based Digital Assistant software.

## B. *Product Result*

Within a period of 4 to 5 months, the author succeeded in making a digital assistant product based on a desktop application. In this version, the application can run 13 commands by typing, there are:

1. Open youtube
2. Search something
3. Find location
4. Play music on youtube
5. What time is it
6. Shutdown my laptop
7. Open google
8. Open gmail

9. Open facebook
10. How are you
11. Wait
12. Nothing/abort/stop/bye
13. Hello

In addition, the Alpha Digital Assistant application can use voice commands. If by using voice commands, users can perform commands other than the 13 commands already mentioned. With voice commands, users can search for news or articles about something with the help of Wikipedia data. Users can also ask questions related to mathematics, science & technology, society & culture, and everyday life with the help of WolframAlpha. This Alpha Digital Assistant application is built using several features that can be created with NLP techniques, including:

a. Text-to-Speech

As the name suggests, this feature will convert text into sound issued by the engine. The text in this application is a string in one of the variables. The string is a command or question that is inputted by the user, either by typing or by voice.

First, the program will initialize the engine provided by the pyttsx3 library and assign a value to the driverName parameter. 'sapi5' for windows and 'nsss' for Mac OS X. Then, the program will adjust the sound of the engine with the getProperty() and setProperty() functions. In the application by default, the voice assistant is a male voice. By running the say() function, the machine can issue a voice assistant through the program.

b. Speech Recognition

If text-to-speech will convert text into sound, then speech recognition will convert the user's spoken voice into text which will be stored in a variable with string data type. It is this variable that will determine what the assistant will do afterwards, as one example has been implemented in figure 5 below.

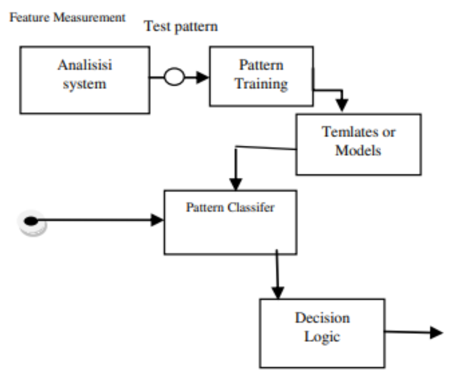


**Fig 5.** Speech Recognition Flow Block Diagram Example

Use the Microphone() function to use the user's computer's default microphone as the program's audio resource. Then the listen() function will be executed to get the user's voice from the audio resource obtained through the user's microphone and then convert it into audio data. The audio data will be recognized by the program using the recognize_google() function so that the audio is converted into string data as a command for the assistant. However, if the audio data is not recognized, an exception error will occur and instruct the user to repeat their speech again.

There are several methods to improve the quality of the microphone so that the program can recognize the user's voice better, but not all of these methods work as I would like. From all the research methods that the author did, the following 2 scenarios were obtained:

1. The quality of the mic is very good so the program can recognize the user's voice very well, but the program is not responding because there is a long process to recognize the voice and the user must speak a few seconds after the assistant invites the user to speak. The following methods are used in this scenario:

    - r.dynamic_energy_threshold = True
    - r.energy_threshold = 4000
    - r.adjust_for_ambient_noise(source)

2. Standard mic quality, but the program can still recognize the user's voice quite well, although it is not suitable if using the program in a very crowded environment. In addition, not responding rarely occurs in this scenario because the speech recognition process is faster than the first scenario, and the user does not need to wait a while to be able to speak after the assistant invites him. The following methods are used in this scenario:

    - r.energy_threshold = 4000

From the 2 scenarios above, the author decided to choose the second scenario because the benefits provided are more than the first scenario. Indeed, the first scenario has better mic quality when used in a crowded environment, but most people use their computers or laptops in a fairly quiet environment, so the drawbacks of the second scenario can be covered.

In addition, when the author uses the first scenario, there is a not responding bug when receiving user voice input for the first time. If the first sound has been recognized by the program, then the next voice input will run smoothly. Even so, this is still very detrimental to the user because of the uncertainty about the smooth running of the program.

So far, the second scenario is going very well after we did repeated testing. The dynamic_energy_threshold and adjust_for_ambient_noise() functions have something in common, namely to adjust the ambient noise level so that the mic can be optimally recognized. It is very suitable for unpredictable noise conditions but has a heavier rendering time so that the program does not respond. Because the application is desktop-based, the noise level is predictable as described previously. Therefore, only the energy threshold function is enough to adjust the noise level so that the program can run more smoothly when recognizing the user's voice. The range of values that can be used include:

- 0 – 100 for quiet environment
- 150 – 3500 for a fairly crowded environment

In addition, it is also possible to provide a higher value. The author uses a value of 4000 because the microphone is sensitive, and the room is quite crowded. With such a high value, the ambient noise can be overcome. In addition, the author has also tried to run the program in a very quiet room and the program can still recognize the user's voice very well.

c. Typo correction

This feature is used when the user gives orders or questions via computer typing. With this feature, users don't have to worry if there is a slight typo in writing the word for the command itself. Researchers have carried out 3 different typo correction methods, including:

1. TextBlob

Only by running the correct() function from this library, the program can correct typo user typing. This method can provide up to 70% accuracy for this spelling correction[21].

However, the drawback is that there are some words that are not in the corpus of this library. One example is the word "google". If you use this method, the word "google" will be changed to "goose" which means goose. This could be because there is no "google" in the English dictionary. While in the digital assistant program there is a command to open google and YouTube, so the author does not use this method.

2. PySpellchecker

This method uses Levenshtein distance to find similarities between 2 words. The author uses the default distance value 2 because the length of the words to be compared is short. However, there is a problem with this method, namely when the program in python file format (.py) is changed to an application file (.exe), the typing command cannot be executed, while when in the python file format (.py) the typing command can be run and running. smoothly.

This is not in line with our expectations that we want to make applications in .exe format so that users can run applications directly without thinking about what requirements must be installed. Thus, the author does not use this method in the program.

3. Jaccard Distance & Unigram

Finally, we use a function from the NLTK library, by combining the Jaccard Distance technique to look for similarities between 1 character or word, with a character unigram to divide the character every 1 value. For example, if the user inputs the question "hiw arr yuu" then this method will perform the following steps:

1) Tokenize the sentence into a list of strings to [“hiw”, “arr”, “yuu”]

2) Finding similar words for each index list, calculating the jaccard distance, and storing it in the list of tuples becomes:

   - “hiw” --> [(0.5, ‘how’), (0.8333333333333334, ‘hello’)]
   - “arr” --> [(0.3333333333333333, ‘are’), (0.6, ‘abort’)]
   - “yuu” --> [(0.3333333333333333, ‘you’), (0.6666666666666666, ‘youtube’)]

3) Change the initial sentence to one of the indexes of the tuple with the smallest distance value and combine them in a new list of strings. So:

   - “hiw” --> ‘how’
   - “arr” --> ‘are’
   - “yuu” --> ‘you’

   List of strings = [‘how’, ‘are’, ‘you’]

4) Concatenate each index of the list into a complete string.

The reason the author uses unigram is because the length of the word is very short, so it is more effective to use unigram. If you use a bigram, let alone a trigram, the result of the distance between "hiw" and "how" and "hiw" and "hello" will be the same, which is 1.0. In addition, the weakness of using this method is that the assistant is only able to use the main features of the application, considering the corpus of the program, the author makes correct words that are used for the main features only.

Therefore, the features of wolframalpha and wikipedia are not used when using the typing command.

d. Knowledge and Answer Engine

Our last feature comes from the WolframAlpha library. This library can only be used if the user gives commands or questions via voice. By using WolframAlpha, users can ask questions from various fields of life, such as mathematics, science & technology, even their daily lives. For example, if the user wants to ask questions related to addition, multiplication, algebra, calculus, and others, the assistant can answer them correctly. Confused about physics or chemistry? Users can also ask the assistant, and there are still many features of WolframAlpha that the author has not explored.

## V. Conclusion

The author has demonstrated making a Digital Assistant, namely Alpha, an intelligent agent product based on a desktop application that performs a task with some specific directions from the user through command instructions either by voice (speech recognition) or typing (text-to-speech). This software evolved from the concept of a virtual personal assistant, a cognitive assistant that can learn and organize. If using voice commands, users can perform commands other than the 13 default commands already available on the main screen of the application. When using voice commands, users can search for news or articles about something, ask questions related to mathematics, science & technology, society, culture, and everyday life with typocorrecting features.

Alpha Digital Assistant combines approaches in artificial intelligence using libraries available in the Python programming language, namely the Pyttsx3, WebBrowser, Random, SpeechRecognition, WolframAlpha, NLTK, Wikipedia, Datetime, Sys, Os, PyWhatkit, Tkinter, and PyInstaller libraries. The author developed this program through a code editor platform, namely Visual Studio Code and the NSIS (Nullsoft Scriptable Install System) installer tools. In addition, to achieve all the existing features, the Alpha Digital Assistant application was built using several features that can be made using natural language processing techniques, namely text to speech, speech recognition, typo correction, and knowledge and answer engine.

The author hopes that the collection of libraries in the Python programming language and natural language processing techniques will develop rapidly in the future so that accessibility from the user's point of view can freely use the digital assistant without any problems that are influenced by the hardware specifications used and the difficulties in adjusting the pronunciation of each word spoken. on voice command instructions (speech recognition).

## References

[1] I. Supiandi, “Analisis Digital Assistant Versi Cortana, Siri dan Google Now,” *Infotech J.*, vol. 1, no. 2, p. 236618, 2015.

[2] P. Bose, A. Malpthak, U. Bansal, and A. Harsola, “Digital assistant for the blind,” in *2017 2nd International Conference for Convergence in Technology (I2CT)*, 2017, pp. 1250–1253, doi: 10.1109/I2CT.2017.8226327.

[3] “Using pyttsx3 — pyttsx3 2.6 documentation.” [Online]. Available: https://pyttsx3.readthedocs.io/en/latest/engine.html.

[4] N. M. Bhat, “pyttsx3 Documentation,” 2021.

[5] “webbrowser — Convenient web-browser controller — Python 3.10.5 documentation.” [Online]. Available: https://docs.python.org/3/library/webbrowser.html.

[6] “random — Generate pseudo-random numbers — Python 3.10.5 documentation.” [Online]. Available: https://docs.python.org/3/library/random.html.

[7] A. Torfi, “SpeechPy - A Library for Speech Processing and Recognition,” *J. Open Source Softw.*, vol. 3, no. 27, p. 749, 2018, doi: 10.21105/joss.00749.

[8] “SpeechRecognition · PyPI.” [Online]. Available: https://pypi.org/project/SpeechRecognition/.

[9] “SpeechRecognition vs speechpy | LibHunt.” [Online]. Available: https://python.libhunt.com/compare-speech_recognition-vs-speechpy.

[10] M. Wickert, “Real-Time Digital Signal Processing Using pyaudio_helper and the ipywidgets,” *Proc. 17th Python Sci. Conf.*, no. Scipy, pp. 91–98, 2018, doi: 10.25080/majora-4af1f417-00e.

[11] “NLTK :: Natural Language Toolkit.” [Online]. Available: https://www.nltk.org/.

[12] “wikipedia · PyPI.” [Online]. Available: https://pypi.org/project/wikipedia/.

[13] “datetime — Basic date and time types — Python 3.10.5 documentation.” [Online]. Available: https://docs.python.org/3/library/datetime.html.

[14] “sys — System-specific parameters and functions — Python 3.10.5 documentation.” [Online]. Available: https://docs.python.org/3/library/sys.html.

[15] “os — Miscellaneous operating system interfaces — Python 3.10.5 documentation.” [Online]. Available: https://docs.python.org/3/library/os.html.

[16] “pywhatkit · PyPI.” [Online]. Available: https://pypi.org/project/pywhatkit/.

[17] D. B. Beniz and A. M. Espindola, “Using Tkinter of python to create graphical user interface (GUI) for scripts in LNLs,” *Proc. 11th Int. Work. Pers. Comput. Part. Accel. Control. PCaPAC 2016*, no. November 2017, pp. 56–58, 2016.

[18] “PyInstaller Manual — PyInstaller 5.1 documentation.” [Online]. Available: https://pyinstaller.org/en/stable/.

[19] J. K. Rask, F. P. Madsen, N. Battle, H. D. Macedo, H. Daniel Macedo, and P. G. Larsen, “Visual Studio Code VDM Support,” no. December, pp. 1–20, 2020,

[Online]. Available: https://pypl.github.io/IDE.html.
[20] "NSIS Users Manual." [Online]. Available: https://nsis.sourceforge.io/Docs/.
[21] "Tutorial: Quickstart — TextBlob 0.16.0 documentation." [Online]. Available: https://textblob.readthedocs.io/en/dev/quickstart.html#spelling-correction.