Remote desktop access from cell phones "Desktop Anywhere"

Bassant Hossam Muhammed
Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt
20201701232@cis.asu.edu.eg

Sara Mazhar Mahmoud
Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt
20201700337@cis.asu.edu.eg

Hossam Hatem Abdel-Baky
Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt
20201701235@cis.asu.edu.eg

Rehab Youssef Hagag
Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt
20201700255@cis.asu.edu.eg

Andrew Samir Barsoum
Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt
20201700156@cis.asu.edu.eg

Dr. Ahmed Salah
Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt

TA. Zeina Rayan
Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo, Egypt

Abstract *— The rapid advancement of technology has significantly impacted our lives, work, and interactions. It has significantly shaped human civilization, making repetitive tasks easier in terms of time, effort, or both.*

Desktop device users face a variety of obstacles while using desktop computers, including Limited Physical Access as Users face the challenge of being physically present at their desktop computers to access or control them, inefficient File and Folder Retrieval in terms of time especially where huge data exists, time-consuming data management across many PCs, also users with disabilities may have difficulty typing, while others may struggle with manual activities such as application launches or shutdowns. Furthermore, individuals who do not properly schedule their work risk becoming overwhelmed, lowering pleasure and productivity.

These obstacles highlight the need for innovative solutions to improve user experience and efficiency. Also, as a response and a contribution to the global orientation, Desktop Anywhere is here. It is a software which developed as a comprehensive and integrated system consisting of two interconnected components: a mobile application and a desktop application communicating through a server. Desktop Anywhere is developed to enable users to access and control multiple personal desktop computers remotely. It aims to provide a seamless experience for users to interact with their desktop screens, manage files, execute tasks through virtual touchpad and keyboard, and utilize voice commands in Arabic, which is done by more than 95% accuracy, all through their mobile devices which will have the ability to be automatically connected to all paired devices without authentication. The primary goal is to bridge the gap between devices and redefine remote interactions, enhancing convenience, efficiency, and accessibility in the realm of digital connectivity.

Keywords—Arabic Language, transformers, BERT, AraBERT, Language Models, Whisper, automatic speech recognition, WebRTC.

I. INTRODUCTION

Desktop device users face a range of challenges and problems that impact their user experience and productivity. These challenges include limited physical access, Users often need to be physically present at their desktop computers, which can be inconvenient and restrict flexibility, inefficient file and folder retrieval as Locating and retrieving files and folders can be time-consuming and frustrating, particularly when dealing with large amounts of data. Time-consuming data management across multiple PCs, difficulties for users with disabilities. Additionally, users who struggle with work scheduling may experience decreased productivity and feel overwhelmed. These problems highlight the need for solutions that enhance user experience, efficiency, and accessibility in the realm of desktop computing.

Thus, the project, "Desktop Anywhere," is a necessary response to the evolving landscape of desktop computing and the challenges faced by users in this field. The rapid advancement of technology has transformed the way we live and work, demanding innovative solutions to enhance user experience and productivity. Within this context, several factors in the field of desktop computing have motivated the idea behind this project.

Firstly, the increasing reliance on desktop computers as essential tools for work and personal tasks has highlighted the need for improved accessibility and flexibility. The project

addresses this need by enabling remote access and control of desktop computers, empowering users to connect and manage their devices from anywhere.

Secondly, the growing complexity of data management has created challenges in locating and retrieving files and folders efficiently. The project aims to streamline these processes, providing users with seamless file and folder retrieval, improving efficiency, and saving valuable time.

Furthermore, the project recognizes the importance of inclusivity and accessibility in the digital realm. Users with disabilities face additional barriers when interacting with desktop devices. By offering virtual touchpad, keyboard, and voice commands, the project caters to the needs of users with disabilities, promoting inclusivity and equal access to technology.

Moreover, the project aligns with the global trend of remote work and digital collaboration. As the world becomes more interconnected, the demand for effective remote access and control solutions has increased significantly. The project embraces this trend, providing users with the ability to seamlessly connect to and manage their desktops remotely, regardless of their physical location.

In conclusion, the project, "Desktop Anywhere," is a necessary and timely response to the challenges faced in the field of desktop computing. The need for improved accessibility, streamlined data management, inclusivity, and remote connectivity has motivated the idea behind this project. By addressing these needs, the project enhances user experience, improves productivity, and aligns with the ongoing advancements in remote work.

II. RELATED WORK

This section provides a comprehensive review of the related work in the remote access software market.

App	Remote Control	Voice Command	Autoconnection	Virtual Touch pad	Multiple Desktops
Team Viewer	✓	X	X	X	X
Any Desk	✓	X	X	X	✓
Chrome Remote Desktop	✓	X	X	X	X
Microsoft Remote Desktop	✓	X	X	X	✓
Splashtop	✓	X	✓	✓	✓
VNC Connect	✓	X	✓	✓	✓

Table 1

Table 1 compares the main features of Desktop Anywhere and the most familiar and known application in today's market in 2024. This comparison highlights the need for the new system.

For Ai based part and models finetuning we will illustrate on the following tables their related work.

Title	Performance	Architecture	Training	Year
Wav2Vec 2.0	Performs well on clean audio data from specific domains (e.g., videos) but struggles with noisy or diverse audio sources. Studies show a Word Error Rate (WER) around 15% on validation sets for clean speech ([Journal of Advances in Information Technology, 2023]). has limitations in cross-lingual tasks and requires fine-tuning with labeled data. generally, shows lower WER on standard benchmarks like LibriSpeech compared to Whisper Wav2Vec 2.0 is highly effective in monolingual speech recognition tasks	Uses a self-supervised learning approach to learn representations from raw audio data	Trained on large unlabeled audio datasets, followed by fine-tuning on labeled data.	2020

Table 2 related work speech to text model before whisper.

Title	Performance	Architecture	Training	Year
Arabic Word2Vec	Generally performs well for simpler classification tasks where word meaning is the primary factor. It can be computationally efficient due to its simpler architecture. Limited effectiveness for NER tasks due to its inability to capture complex relationships between words. **1.Limited context:** Doesn't capture the relationships between words. **2.Limited vocabulary** **3.Static representations:** Word embeddings are fixed and don't adapt to specific tasks.	two-layered neural network architecture that excels at learning word embeddings 1. Input layer 2.Hidden layer 3.Output layer	Two training approaches exist within Word2vec 1.Continuous Bag-of-Words (CBOW) 2.Skip-gram	2013

hULMonA: A Universal Language Model for Arabic	While hULMonA shows promise for Arabic text classification tasks, there are some limitations to consider: **1.Limited transparency:** The full details of the model architecture haven't been publicly released. **2.Data dependency:** hULMonA's performance relies heavily on the quality and size of the training datasets. If the fine-tuning data is limited or not representative of the target task, accuracy might suffer. **3.Potential bias:** Like many NLP models, hULMonA could inherit biases present in the training data.	The exact architecture details haven't been publicly disclosed	hULMonA employs a three-stage training process: 1.Pre-training 2.Fine-tuning 3.Classification	**2019** (Paper: [ElJundi et al., 2019])
Multilingual BERT (mBERT)	Generally performs well for basic tasks like sentence classification, but its accuracy might be lower than AraBERT for tasks requiring a deeper grasp of Arabic.	leverage the Transformer architecture	Trained with a Masked Language Modeling (MLM) objective. The model predicts masked words based on surrounding context, allowing it to learn relationships between words across languages.	2018 (Paper: [Devlin et al., 2018])
T5 (Text-to-Text Transfer Transformer)	May perform adequately for some basic Arabic tasks after fine-tuning, but its accuracy might be lower than AraBERT v2 for tasks requiring a deeper understanding of Arabic	Encoder-decoder architecture.	Trained using a "masked language modeling" (MLM) objective along with a variety of other text-to-text tasks in a multi-task learning approach	2019 (Paper: [Raffel et al., 2019])

Table 3 compares some models used for classification and NER tasks for Arabic and Multi-language.

III. PROPOSED METHOD

In our software project, we have implemented several key modules. The project consists of the following modules: Server, Mobile App, Desktop App, Model Finetuning, and Dataset Creation.

Server

The server is responsible for handling communication between the mobile and desktop applications. It was developed using NodeJS and ExpressJS to manage real-time data transmission and ensure seamless interaction between client applications. The server acts as a central hub, facilitating the exchange of data and commands between the mobile and desktop environments.

Mobile App

The mobile app provides a user-friendly interface for voice command entry and real-time interactions. It was developed using the Dart programming language and the Flutter framework. For real-time view features, we integrated WebRTC, an open-source technology that enables the transmission of sound, video, and data over the internet. This integration allows for real-time communication across different devices, enhancing the user experience.

Desktop App

The desktop app allows users to interact with the system via voice commands on a desktop environment. It was developed using Python and communicates with the server to execute commands and retrieve data. The desktop app provides a seamless and efficient way for users to
manage their tasks through voice commands. Additionally, users can remotely access and control the desktop or several desktops using the mobile application, offering comprehensive remote management capabilities.

Model Finetuning

We utilized two AI models for voice command understanding: Whisper and AraBERT. Whisper is a transformer-based model designed for speech-to-text conversion, while AraBERT is used for classification and Named Entity Recognition (NER) tasks. The process begins when a user initiates a voice command using the mobile app. The audio input is processed by the Whisper model, which generates a textual transcription of the user's voice command. The transcribed text is then classified using the AraBERT model to determine the type of command issued by the user. Possible command categories include search, set, add note, close, restart, and sleep.
Once the command type is identified, AraBERT performs NER to extract useful information from the command, such as file names, folder names, time, and other relevant entities. After extracting these entities and understanding the command context, the appropriate action is executed on the desktop app.

Dataset Creation

To finetune the Whisper and AraBERT models, we created a customized dataset in two forms: text and voice records. The dataset covers the six main commands allowed in our system for users to ask verbally. The steps in the dataset creation were as follows:

1. ***Text Data Creation****: We created 1872 rows of text data.*
2. ***Voice Record Collection****: We found volunteers who each recorded an average of 20 sentences, resulting in 1932 voice records. The difference in the number of records is due to the need for balance in the text dataset, which affects classification and NER tasks.*

This comprehensive dataset was then used for finetuning the models.

IV. DATASETS

This paper incorporates two distinct datasets to facilitate the training and evaluation of the two models implemented within this project. Each dataset is described below, providing information regarding the data format, number of entries, and other relevant details.

Before we go deeper in describing our dataset and its creation and evolution phases let's illustrate the Permitted Voice commands through our system. Users can:

- ***Search****: The user can search for a file/folder on his desktop.*
- ***Set*** *: The user can set the alarm or the stopwatch on his desktop device.*
- ***Add Note*** *: The user can add note on his desktop device.*
- ***Close*** *: The user can schedule shuting down the desktop or closing specific running app.*
- ***Restart*** *: The user can schedule restarting the desktop.*
- ***Sleep*** *: The user can schedule putting the desktop on sleep mode.*

Our initial task proved to be a challenge. We looked for a collection of data (dataset) containing this type of command on websites like Kaggle, but unfortunately, we couldn't find one. There wasn't a real-world situation (context) where this kind of command is commonly used, so we couldn't easily collect data from one. Since there wasn't any existing data available, we had to create our own dataset by generating the data ourselves. Before we go for start creating the data we analyzed the needs of each model. Whisper needs: to fine tune whisper we want commands in form of voice records, Arabert needs: to fine tune Arabert we want commands in a text form.

After several trials and errors, we reached our goal, and the following figure shows a dataset sample from final version of the dataset which is used in AraBERT finetuning and then recorded to use these records to finetune Whisper. In the following figures we will show the final size of our data set and its details.

	A	B	C
1	sentence	class	label
2	ابحث عن مجلد image processing	بحث	{"اسم العنصر":"image processing","نوع العنصر":"مجلد"}
3	ابحث في الحاسوب عن مجلد grades	بحث	{"اسم العنصر":"grades","نوع العنصر":"مجلد"}
4	جد لي مجلد حسابات علمية	بحث	{"اسم العنصر":"حسابات علمية","نوع العنصر":"مجلد"}
5	في تمام 10 قم بغلق الجهاز	غلق	{"اسم العنصر":"الجهاز","وقت":"10"}
6	انتظر حتى الساعة 13 ثم اوقف تشغيل الجهاز	غلق	{"اسم العنصر":"الجهاز","وقت":"الساعة 13"}
7	بعد نصف ساعة اغلق الجهاز	غلق	{"اسم العنصر":"الجهاز","وقت":"نصف ساعة"}
8	بعد الساعة 13 قم بالغلق برنامج blender	غلق	{"اسم العنصر":"blender","وقت":"الساعة 13"}
9	انتظر ساعة ثم اعد تشغيل الجهاز	اعادة تشغيل الجهاز	{"اسم العنصر":"الجهاز","وقت":"ساعة"}
10	انتظر 15 دقيقة ثم قم باعادة تشغيل الجهاز	اعادة تشغيل الجهاز	{"اسم العنصر":"الجهاز","وقت":"15 دقيقة"}
11	بمرور 4 ساعات قم باعادة تشغيل الجهاز	اعادة تشغيل الجهاز	{"اسم العنصر":"الجهاز","وقت":"4 ساعات"}
12	اضبط المنبه ليتبهني بعد الساعة 24	ضبط	{"وقت":"الساعة 24","اسم العنصر":"المنبه"}
13	اضبط المنبه ليتبهني بعد مرور ساعة كاملة	ضبط	{"وقت":"ساعة","اسم العنصر":"المنبه"}
14	اضبط المؤقت ليتبهني بعد 35 دقيقة	ضبط	{"وقت":"35 دقيقة","اسم العنصر":"المؤقت"}
15	اضبط المؤقت ليتبهني بعد 40 دقيقة	ضبط	{"وقت":"40 دقيقة","اسم العنصر":"المؤقت"}
16	قم بضبط المؤقت لمدة 50 دقيقة	ضبط	{"وقت":"50 دقيقة","اسم العنصر":"المؤقت"}
17	قم بتعيين المؤقت لمدة 65 ساعة و17 دقيقة	ضبط	{"وقت":"65 ساعة و17 دقيقة","اسم العنصر":"المؤقت"}
18	بمرور ثلث ساعة قم بوضع الجهاز في وضع السكون	ايقاف تشغيل الجهاز مؤقتاً	{"وقت":"ثلث ساعة"}
19	بمرور 6 ساعات قم بوضع الجهاز في وضع السبات	ايقاف تشغيل الجهاز مؤقتاً	{"وقت":"6 ساعات"}
20	عد ساعتين وبيع لم قم بايقاف تشغيل الجهاز مؤقتاً	ايقاف تشغيل الجهاز مؤقتاً	{"وقت":"ساعتين وبيع"}
21	اكتب في مذكراتي الآتي علينا قراءة النص بعناية قبل نشره	اضافة ملاحظة	{"الملاحظة":"علينا قراءة النص بعناية قبل نشره"}
22	اضف الي ملاحظاتي لدي موعد عشاء في نفس يوم الاختبار	اضافة ملاحظة	{"الملاحظة":"لدي موعد عشاء في نفس يوم الاختبار"}
23	قم باضافة الجملة التالية الي مذكراتي كان الاجتماع مثمر وتم تغطية جميع النقاط المهمة	اضافة ملاحظة	{" الملاحظة":"كان الاجتماع مثمر وتم تغطية جميع النقاط المهمة"}

Figure 1

Description of final version of Dataset

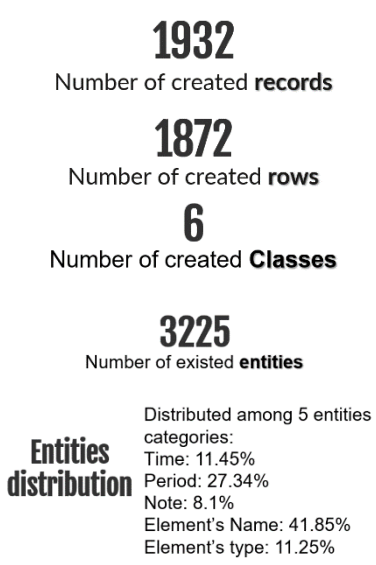

Figure 2

V. RESULTS & DESCUSSION

Expected Results:

1.1. Collecting Voice Dataset for Whisper.
1.2. Collecting textual Dataset for Multi-class classification in Arabic for AraBERT.
1.3. Collecting textual Dataset for NER task in Arabic for AraBERT.
1.4. Finetune them for the following tasks speech to text, classification and Name Entity Recognition to reach our goal which is getting accuracy 95% or higher.
1.5. Building Mobile app, Desktop app and server.
1.6. Deploy our server on free online server.
1.7. Building Command execution system.

Actual Results:

1.1 Voice Dataset collected.
1.2 Dataset for Multi class classification collected.
1.3 Dataset for NER Arabic collected.
1.4 Models for speech to text, classification and Name Entity Recognition has been chosen and finetuned depends on the following trials.
1.5 Mobile app, Desktop app and server built and the Server deployed.
1.6 Command execution system built.

Final results after a lot of trials for ai based part:

1 Speech to Text Task: we chose Whisper model for this task we finetuned it with 1932 record which met our targeted accuracy, and the results we got from final trail is displayed in table 2.
2 Results of finetuning Ara-BERT for Classification task mentioned in table 3.
3 Results of finetuning Ara-BERT for NER task mentioned in table 4.

	Customized Dataset
SIZE	***1932***
SPLITS	***Train: 1544*** ***Test: 387***
WER Before Fine-tuning	***28 %***
WER After Fine-tuning	***4.6%***

Table 2

	AraBERT
SIZE	***1872***
SPLITS	***Train : 1497*** ***Test : 375***
Accuracy Before Fine-Tuning	***25%***
Accuracy Before Fine-Tuning	***100%***

Table 3

	AraBERT
SIZE	***1872***
SPLITS	***Train: 1497*** ***Test: 375***
Accuracy Before Fine-Tuning	***43 %***
Accuracy Before Fine-Tuning	***98 %***

Table 4

VI. CONCLUSION

In conclusion, in this project we tried to bridge the gap between physical presence and desktop accessibility. Through Desktop Anywhere, a comprehensive system designed for users to be user-friendly, the mobile application empowers users to access and control their personal desktops from anywhere through the server. This remote access capability simplifies file management by allowing users to effortlessly manage and navigate through files and resources across multiple desktops concurrently.

Desktop Anywhere embraces inclusivity by providing accessibility features like voice commands (currently in Arabic) that cater to a wider range of users. For individuals who prefer hands-free interaction or those engaged in multitasking activities, Desktop Anywhere provides voice control and virtual input methods for executing tasks and controlling applications. Repetitive tasks like shutdowns or application launches can be automated, streamlining daily routines and boosting efficiency.

The ability to manage tasks effectively is crucial in today's fast-paced world. Desktop Anywhere addresses this need by offering task scheduling, allowing users to organize activities seamlessly across both desktop and mobile devices.

ACKNOWLEDGMENT

All praise and thanks to ALLAH, who provided us with the ability to complete this work. We hope to accept this work from us. Also we are grateful to our family, especially our parents who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

We also offer our sincerest gratitude to our supervisors, Dr. Ahmed Salah, T.A. Zeina Rayan, who have supported us throughout our thesis with their patience, knowledge, and experience. Finally, we would thank our technical mentors Eng. Manar Sultan and Eng. Reem Telbani who gave us support and encouragement.

REFERENCES

[1] *A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," arXiv:2212.04356 [cs, eess], Dec. 2022, Available: https://arxiv.org/abs/2212.04356*

[2] *Abdelhalim Hafedh Dahou, Mohamed Amine Cheragui, and A. Abdelali, "Performance Analysis of Arabic Pre-Trained Models on Named Entity Recognition Task," Jan. 2023, doi: https://doi.org/10.26615/978-954-452-092-2_051.*

[3] *W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," arXiv:2003.00104 [cs], Mar. 2020, Available: https://arxiv.org/abs/2003.00104*

[4] *WebRTC, "WebRTC Home | WebRTC," Webrtc.org, 2017. https://webrtc.org/*

[5] *V. Tiwari, M. F. Hashmi, A. Keskar, and N. C. Shivaprakash, "Virtual home assistant for voice based controlling and scheduling with short speech speaker identification," Multimedia Tools and Applications, Jul. 2018, doi: https://doi.org/10.1007/s11042-018-6358-x.*

[6] *F.-C. Cheng, "Automatic and Secure Wi-Fi Connection Mechanisms for IoT End-Devices and Gateways," Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 98–106, 2018, doi: https://doi.org/10.1007/978-3-319-95450-9_8.*

[7] *"DistilBERT,"huggingface.co.https://huggingface.co/docs/transformers/model_doc/distilbert#usage-tips*