Intelligent Knowledge Bank on AR Platform with Open-World and Metaverse Compatibility

Zeyu Liu[1*], Ruizhi Qian[2], Boxuan Shang[3], Kezhen Zhang[4]

zliu0251@uni.sydney.edu.au[1*], qrz1999526@gmail.com[2], Shawn070112@outlook.com[3], xvwj53@durham.ac.uk[4]

[1] School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
[2] School of Energy Resource, China University of Geosciences (Beijing), Beijing, 100083, China
[3] Keystone Academy, Beijing, 101318, China
[4] Department of Computer Science, Durham University, Durham, DH1 3LE, England

Abstract: This paper provides a solution for the forgetful problem that we seek in Alzheimer's patients and extends the solution to meet the need of all members of society. This paper is split into 7 parts, including a short introduction of what is the problem and background information about the context, some related work similar to the product we are creating; a detailed solution design that will solve the problem with consideration made very carefully; the solution implementation part will introduce the minimal viable prototype (MVP) we build so far to demonstrate our product; the lesson learned and next step section would introduce some knowledge we learnt during this project and how it can be used in a larger aspect of view; the conclusion would sum up all the points and last but not least, the reference section which include all sources we have been looking to gain ideas and inspirations. The solution we suggest is a cloud-based software, using technologies such as AI, Machine learning, Augmented reality, data storage, etc. The software will be a knowledge bank system designed to help people "remember" important things and remind them using intuitive UI widgets in the AR hardware. The forgetful problem we came up with is an actual problem existing across our society, and the problem-solving technique is specific and challenging because of the involvement of different technologies, both modern ones and near-future ones.

Keywords: Knowledge Bank, Augmented Reality, Open World, Software-as-a-Service, Metaverse

1. INTRODUCTION

This idea of creating this knowledge management system origin from the stimulation of a society's status, the growing community of Alzheimer's disease have become a severe problem that governments are making significant efforts to ameliorate the situation. When COVID-19 became "popular", the elder patients suffered as governments decided to put more effort into the younger population on this new pandemic and had to reduce support for the elders. Our product is not designed to become a medical device that physically helps the patient, but designed as software, building up a knowledge bank, supporting mental illness and people who have this need to remember a bunch of things. The original idea was to create a medical guidance system for the Alzheimer's group of people to recognize the faces of people they know and distinguish

EMIS 2023, February 24-26, Chengdu, People's Republic of China

DOI 10.4108/eai.24-2-2023.2330624

them from strangers so that wouldn't suffer from being tricked by bad people. The product of idea 1.0 would be limited to only Alzheimer's patients. The team further researched this idea and decided to expand the possibility of further development of this idea by turning it into a knowledge bank software that not only has the ability to help forgetful people but is also capable of becoming an open-world platform that is open to be implemented into all kinds of environments, including in Metaverse, and be helpful to the whole society. By making its knowledge bank, we now have the possibility of keeping track of people's experiences and might further develop it to become a digitized medium of knowledge that can be stored and passed along without loss of details.

2. RELATED WORK AND LITERATURE REVIEW

2.1. Disaggregated Cloud Memory with Elastic Block Management

In life and work, there are more and more data that need to be stored, transmitted, or shared. Cloud storage services emerged as the volume of data is growing at an explosive rate.

With the increasing requirement for memory size by memory-intensive workloads, cloud service providers have launched large memory virtual machine (VM) services to satisfy the demand of the market. Large memory VM services use small capacity expansion machines at a lower cost than an expansion service consisting of commercial servers using large memory. By taking advantage of the imbalance of memory usage between cloud nodes, the decomposed memory can be used to expand the memory capacity of virtual machines in a cost-effective way to meet the in-memory data processing requirements. With cloud service providers' memory paging technique, a single large memory block can be decomposed into multiple pieces to be distributed to different remote nodes while operating as un-modified. By doing so, programs with high memory demand can both run faster by utilizing more direct physical memory and run smoother by meeting the large memory capacity requirements [1, 6]. Kwangwon shows a new method to enlarge the cloud storage, which is to build a hypervisor-integrated disaggregated memory system for the cloud storage. With Kwangwon's team's solution, even if the direct memory capacity is only 50% of the total memory footprint compared to a machine that uses only local memory. The split memory provides an average performance degradation of only 6%, which means that the user can achieve the goal with half the original memory. Our Intelligent Knowledge Bank is a cloud-based application, and we might need thousands of gigabytes of space to store the data when the user base grows. Large cloud storage would become necessary.

The method provided by Kwangwon's team could help improve both developers' experience of optimizing the application and the user experience of having a smooth cloud application without worrying about the cost of high-capacity cloud storage.

2.2. AR Map Dynamic Navigation System

This paper discloses an AR map dynamic navigation system. The system creates a virtual scenic map generated by the specific real-time information of the scenic spot and the information of the path to the scenic spot. A dynamic image is generated on mobile terminals and constant tracking and displaying the current location. At the same time, the system generates the information of virtual tourists according to the real-time location information and marks the location of tourists on the map. When the desired scenic spot location is selected on the map, a path is generated to navigate the scenic spot, and the path is displayed on the virtual scenic spot map. And the virtual scenic spot map would be projected onto the display device of the mobile terminal through augmented reality devices [2, 10]. This article publishes a new type of AR map which use the real-time position and tourist to generate a model, and then put these into the terminal to be recorded. The solution described in this article can be integrated into our project as further development. With this new feature, we could help people remember the locations and guide the user to the locations. Also, by combining the power of IoT, we might be able to guide users to their lost daily items.

2.3. Software-as-a-Service

Software-as-a-Service, also known as SaaS, means a server provides software services over the network. Also, SaaS is a new software delivery model built in a service-oriented way. At the same time, SaaS and service-oriented software are different [House, 2009]. SaaS emphasizes building customizable architectures and large-scale scalable systems on top of the cloud infrastructure. SaaS supports different levels of customization and scalability through database and design techniques, which include user interfaces, workflows, and persistence. In this way, thousands of tenants can use the same SaaS infrastructure with the same service database at the same time. By applying different configurations, tenants could use the same structure as a new dedicated application.[7-8] By building our solution on the cloud, we would not need to purchase and maintain any hardware or server infrastructure.

3. SOLUTION DESIGN

Our solution to address the problem stated in the introduction is to create software that would be able to recognize, and remember identities and conversations of people and display the information to the user when needed using, preferably, AR. To adapt to deploying the solution in the open world today and the coming metaverse, our solution would be distributed on the cloud and have the capability to change the way it behaves in different environments. Also, despite we designed this application to be a helpful tool, it can be easily turned into a profitable product paying with both crypto-or-regular currency.

The features that our solution has can be summarized in a scenario description. When the user meets some person X, the application would first try to recognize X by matching X with existing identities in the database. If X's information is in the system, X's detail would be displayed in a widget floating next to the person in the user's AR device, also if any conversation has been made with X before would be displayed in another widget floating next to X. If there's no information about X in the database, the user could choose to add X into the user's personal database and keep records of the conversation between them.

The scenario above shows the general functionality of the application, but in our design, it can be adapted to the environment of its deployment. For example, if the application is deployed within a scope of a single company. All employees' information should already exist in the database, and the ability of new identity creation would be extracted from regular users and managed by the administration of the company. And information stored and displayed can be not only conversations between people but also data about an ongoing project.

To accomplish the feature we state above, we need to implement the following modules: main program and UI, face detection and recognition, voice record and content extraction, knowledge management system, and UI. The architecture of the solution is demonstrated in Fig.1 as follows:

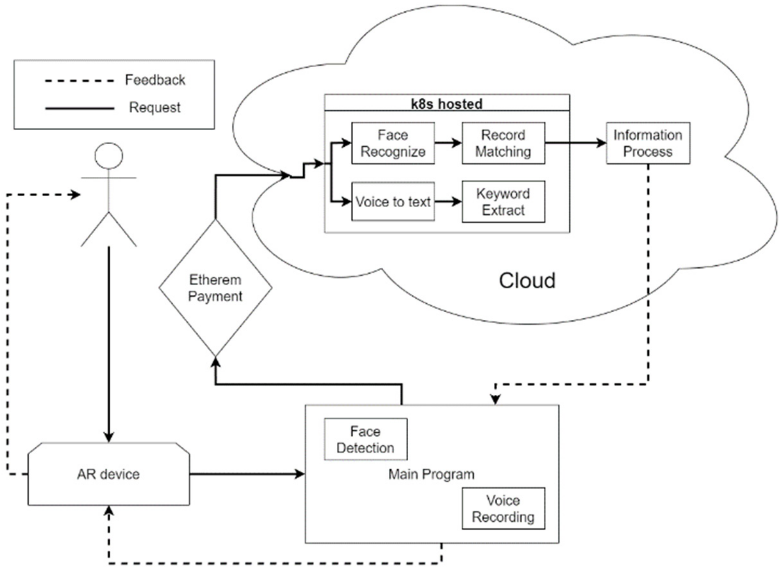

Figure 1. The structure of our solution

As most of the features are quite computing-intensive, those features would be handled on the cloud to utilize the scalability of cloud deployment tools like Kubernetes and the computing power of cloud services. The main program and UI should be lightweight in terms of computing and much closer to the end user and would be deployed locally on the user's AR device.

3.1. Main Program and UI

This module would act as an entry part of our solution, gathering the functionality of other modules into one service. The user interface in our solution should be implemented in AR as widgets floating around our user and next to people the user is meeting with. Operating according to a user-specified configuration, the program can perform local tasks as the wish of the user, send some computing-intensive work to the cloud for faster processing and display the result that may interest the user conveniently and intuitively.

3.2. Face Detection and Recognition

This module would be used to detect and recognize faces for identity checks and verification. This module is tricky to design as both functionalities are intensive in terms of computing as both require graphic/video processing. In the pre-5G era, it is unlikely that we can have low-latency high-quality video transmission for sending visual data to the cloud for analysis, as the 4G network's speed is capped at 128 Mbit/s [9]. Therefore, we choose to put face detection as a local feature, it would detect existing faces in front of the AR device, crop out the faces and sent the faces to the cloud servers. On the cloud, the faces' features would be extracted utilizing a trained deep learning model and compared with existing face data to figure out the identity of the faces.

3.3. Voice Record and Content Extraction

This module would be used to record live communication between the user and other people and extract the keywords from the conversations for memory storage. This module consists of three components: voice recording, voice-to-text conversion, and conversation content extraction. The voice recording component would be built into the main local program as it requires physical presenting devices for recording the conversation. The voice record would then be sent to the server on the cloud for other heavy-in-computing tasks. The next task is to convert the voice record to readable text for convenience in the later content-extract process. This task could be achieved by using existing voice-to-text Application Program Interface (API) from various cloud services like the Speech-to-Text service from Google Cloud Platform [3]. After getting the text of the conversations, we figured by default the solution should only extract and store the keywords of conversations into the database, as the volume of conversations is diverse and the size of data could become unpredictable. By extracting keywords with AI-based cloud services, we can store the main idea of the conversation for users to look up the related conversation faster and allows building a timeline for the same topic in case of demand.

3.4. Knowledge Management System

This module would be used to keep and maintain the data of stored user information, including facial features, conversation keywords, and more. This module would be entirely deployed on cloud platforms for two significant reasons. The first one is that most of our modules are running on the cloud, having the databases on the cloud would save the communication delay between the cloud and the end user. The other reason would be by utilizing cloud storage, the solution's data management system could use the existing cloud storage system with high scalability, high resilience and less possibility of data loss. The structure we use to build our system would be consistent with both SQL and NoSQL database systems. SQL databases would be used to store relational information like identities and relative data, NoSQL part would be images of people and most importantly the facial feature of individuals. And can be further improved by integrating both parts using the around-corner NewSQL system.

With the structure above, we can also have a similar kind of knowledge bank implemented in the Metaverse environment. It can be achieved by adding another module that records other people's Metaverse identity and, at the user's and others will, maybe matches to existing physical-world identity. The main purpose of the solution in the real world can be remained, as a general conversation diary or other extended applications. Also, the payment system can be built with Ethereum as the project implemented as a product in Metaverse. Further development and improvement will be discussed in Chapter V below.

4. SOLUTION IMPLEMENTATION

In a relatively short time, the team could not build an entire system that utilizes all technology mentioned above, but we were able to build a minimum viable prototype (MVP) in python that implements the core modules required to fulfil our functionalities.

The python MVP we developed consists of 5 python scripts:

4.1. __init__.py

This python script is used as a text user interface entrance for our MVP, it could be used to call functions from other modules to achieve the ability to get facial feature data, store identity information, face detection and recognition, voice recording and keyword extract function and other background data management functions.

4.2. database_operations.py

This python script is acting as the knowledge management system module of our solution but is fully implemented locally. This module uses Python's built-in sqlite3 library to build and manage the SQL database of our solution. The SQL database in the MVP consists of four tables:

1) Table People:

This table stores information like identity id, name of identity, age, and the relationship between the identity and the user.

2) Table Face

This table stores the link between people's identities with facial feature NoSQL data.

3) Table Context

This table stores the key content and other metadata of a conversation.

4) Table Conversation

This table stores the link between people's identities with conversation information.

The NoSQL part of the database system has two parts of content, the first part is the facial feature files dumped from Python objects generated in face_recog module below using numpy. save, and the other part is the corresponding facial images of the facial features for human-perceptible management.

4.3. face_recog.py

This python script is acting as the face detection and recognition module of our solution but is fully implemented locally, while also temporarily featuring a "like-AR" widget that displays relative information next to the face recognized. Both face detector and recognizer use OpenCV's built-in structure FaceDetectorYN and FaceRecognizerSF with the power enhanced by power deep-learning-trained models, YuNet and SFace, from the opencv_zoo repository [4]. This module, working with the database_operations module, would allow face detection, facial feature capture and store, facial feature and identity matching, face recognition and data displaying.

4.4. speech.py

This python script is not integrated into the application as in MVP, but it represents the way we can utilize the cloud services to accomplish the features of our solution. This script represents the voice record and content extraction module of our solution. The script has the ability to record voice, send it to Baidu's voice-to-text cloud service to get the text of the conversation, and then use BERT model to extract the keywords of the conversation for index usage.

4.5. util.py

This python file is a simple library for frequently used methods across all other modules.

The URL of the repository of this MVP on GitHub is: https://github.com/LeMonThELord/ns-summer-2022-project

5. LESSON LEARNED AND FURTHER DEVELOPMENT

During the process of developing our minimal viable prototype, we have learned that making use of cloud technologies can significantly accelerate our development progress and improve our application. For example, in the part of voice recognition, we utilize a SaaS from Baidu Cloud. It provides us with a real-time voice recognition service that meets our needs. As a result, we do not need to build such a program from scratch (as the picture shows, it has multiple APIs supporting different languages). In other parts of our application, we can also exploit such cloud services, which can reduce the amount of work in the development phase and facilitate the management of our application.

Because our application involves a lot of human-computer interaction, we should devise it from a user's perspective, considering what our users need and making it as convenient as possible. For example, in the part of keyword extraction, we should consider the number of keywords for different pieces of text. Generally speaking, the longer the text is, the more keywords we need to generate. However, each piece of text should have at least three keywords because it is difficult for ordinary people to remember anything with only one or two keywords even if the text is very short. For Alzheimer's patients, the number of keywords should be quite larger.

Latency is the main barrier to our application being enough immersive especially when we transfer some parts of our application to the cloud. In human-computer interaction, high latency makes our application seem clumsy. 5G and edge computing could be a feasible solution to this problem so it may be necessary for us to leverage these technologies in further development.

The further development phase is comprised of four steps and each step paves the way to the next step.

5.1. Polishing MVP

Firstly, the whole application should be containerized by Docker and deployed on the cloud. The whole database should be transferred to the cloud as well.

Secondly, a user interface that can show the content of the conversation and some keywords extracted from it should be created. Thus, after the conversation, the users can review the conversation through the headset and do some basic management on the User interface, such as adding some paraphrases. In addition, the user interface should be compatible with further development.

Thirdly, in order to lower the latency and meet its compute-intensive workload, the application should be combined with 5G and edge computing technologies such as Multi-access Edge Compute (MEC).

Finally, some real patients or other people who are interested in our project are invited to do some tests.

5.2. Leveraging IoT and Blockchain Technology

The application will have a new function by implementing IoT technology. This new function enables our users to manage the locations of many daily items such as the locations of their cars and smartphones, and the mixed reality devices can visualize these locations and guide our users.

In terms of blockchain, our application can be paid using Ethereum for purchasing the software or subscription. In addition, we may use smart contracts to manage some crucial information. Now, some people may be afraid that a knowledge bank on the cloud could be insecure. If some records in a knowledge bank are falsified, it will bring much trouble for our users. It is essential for our users to record some crucial information such as the name of important business partners and the conversation recording of this person on the blockchain.

5.3. Adapting our Application to the Metaverse

A knowledge bank can also be very useful in the virtual world. Meta-verse represents the future of the internet. It is important for us to adapt our application to the meta-verse. The knowledge bank will be not only able to store and manage the information about people in the real world, but also be able to identify virtual identities in the meta-verse and record information associated with these virtual identities.

To adapt our application to the realm of meta-verse, we need to do the following things:

(a) The idea of meta-machine learning should be implemented to Improve our facial and voice recognition algorithms. Because we must figure out whether our users are in meta-verse or in the real world and choose the corresponding functions. Besides, in the future, there will be many kinds of meta-verse from different companies. Each of them may have unique features (for example, they may have different styles of avatars). Thus, it is crucial for us to implement the idea of meta-machine learning that enables us to do facial recognition (in the meta-verse it would be avatar identification) precisely.

(b) We need to create a database to record the information associated with this virtual identity. Different kinds of meta-verse need to have different databases. Those databases should be parallel with each other to avoid any confusion.

(c) Finally, we need to enable our users to relate virtual identities in different kinds of metaverse to the people in the real world.

5.4. Adding more Functions Based on the Knowledge Bank

First, we will enable our users to share or sell their own knowledge banks to others. A possible application scenario is the training for salesmen. Salesmen need to communicate with lots of people every day. Talking to people in a proper way is essential for this job. Some new salesmen may want to buy the knowledge storage from those successful salesmen to improve their communication skills

Second, the knowledge bank could provide a Huge conversation database for the development of chatbots. We have connected the conversation data to different individuals. So, we can also develop chatbots talking in a specific way. We can use the information from children to make a chatbot chatting like a child. Such a kind of chatbot would be very helpful for those parents who lost their children.

Third, the knowledge bank also includes information about virtual identities in the meta-verse. We can generate lots of bots in a virtual world, these bots can talk with users like true humans, which helps the meta-verse to be more immersive.

6. CONCLUSION

Our intelligent knowledge bank is based on cloud technologies, artificial intelligence, and AR/VR technologies, which were developed to help people recognize, remember and manage different people's identities and conversations. It can be adapted to the environment of its deployment and different kinds of Metaverse in the future.

The whole application is comprised of the following modules: main program and UI, face detection and recognition, voice record and content extraction, and knowledge management system. The identities of different people can be recognized by the face detection and recognition module. The content of conversations can be recorded, and some keywords of the conversations can be generated by the voice record and content extraction module. The knowledge management module maintains the data from them, and the UI can visualize the information the users need through augmented reality. In addition, the UI can also enable users to do some management. Consequently, our users can see the information of other people (identities, conversation keywords before, etc.) in a widget in their AR device, and the conversation between our users and other people can be recorded, analyzed, and displayed in real-time simultaneously.

Our application makes full use of several cloud technologies to accommodate our database and computing workload. In the future, we plan to implement 5G technologies and edge computing to reduce latency.

Now, we have created an MVP of our application. Our MVP successfully demonstrated several core functions of our product including facial recognition, voice recording, keyword extraction, and basic knowledge management. It proves the feasibility and the functionality of our application.

Our application also opens a door to innovations based on a knowledge bank. Users of such a knowledge bank can sell their knowledge to others or purchase an expert knowledge bank to improve themselves. Developers can take advantage of the huge amount of data in the knowledge bank to build a better chatbot or generate some virtual identities in a metaverse.

REFERENCES

[1] Chen S , Wang W , Wu X , et al. Optimizing Performance and Computing Resource Management of In-memory Big Data Analytics with Disaggregated Persistent Memory[C]// 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). ACM, 2019.
[2] D. Zhou et al., "AR map dynamic navigation system", CN110609883A, 2022.
[3] Google Cloud. "Speech-to-Text: Automatic Speech Recognition | Google Cloud", 2022. [Online]. Available: https://cloud.google.com/speech-to-text/. [Accessed: 12- Aug- 2022].
[4] GitHub. "GitHub - opencv/opencv_zoo: Model Zoo For OpenCV DNN and Benchmarks.", 2022. [Online]. Available: https://github.com/opencv/opencv_zoo. [Accessed: 12- Aug- 2022].
[5] House M . "Configurability in SaaS (software as a service) applications", in ISEC '09: India Software Engineering Conference, Pune India, 2009.
[6] K. Koh, K. Kim, S. Jeon and J. Huh, "Disaggregated Cloud Memory with Elastic Block Management", IEEE Transactions on Computers, vol. 68, no. 1, pp. 39-52, 2019. Available: 10.1109/tc.2018.2851565.
[7] W. Tsai, X. Bai and Y. Huang, "Software-as-a-service (SaaS): perspectives and challenges", Science China Information Sciences, vol. 57, no. 5, pp. 1-15, 2014. Available: 10.1007/s11432-013-5050-z.
[8] W. Tsai, Q. Shao and W. Li, "OIC: Ontology-based intelligent customization framework for SaaS", in 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), 2010, pp. 1-8.
[9] Wikipedia. "4G - Wikipedia", En.wikipedia.org, 2022. [Online]. Available: https://en.wikipedia.org/wiki/4G. [Accessed: 12- Aug- 2022].
[10] Yuan Y , Zhu M , Jiang Y , et al. A bare hand controlled AR map navigation system[C]// IEEE International Conference on Systems. IEEE, 2005.