Interactive Cycle Model: The Linkage Combination among Automatic Speech Recognition, Large Language Models, and Smart Glasses

Libo Wang
Nicolaus Copernicus University
Jurija Gagarina 11, 87-100 Toruń, Poland
326360@o365.stud.umk.pl
UCSI University
Taman Connaught, 56000 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur, Malaysia
1002265630@ucsi.university.edu.my

Abstract

This research proposes the interaction loop model "ASR-LLMs-Smart Glasses", which model combines automatic speech recognition, large language model and smart glasses to facilitate seamless human-computer interaction. And the methodology of this research involves decomposing the interaction process into different stages and elements. Speech is captured and processed by ASR, then analyzed and interpreted by LLMs. The results are then transmitted to smart glasses for display. The feedback loop is complete when the user interacts with the displayed data. Mathematical formulas are used to quantify the performance of the model that revolves around core evaluation points: accuracy, coherence, and latency during ASR speech-to-text conversion. The research results are provided theoretically to test and evaluate the feasibility and performance of the model. Detailed architectural details and experimental process have been uploaded to Github, the link is:https://github.com/brucewang123456789/GeniusTrail.git.

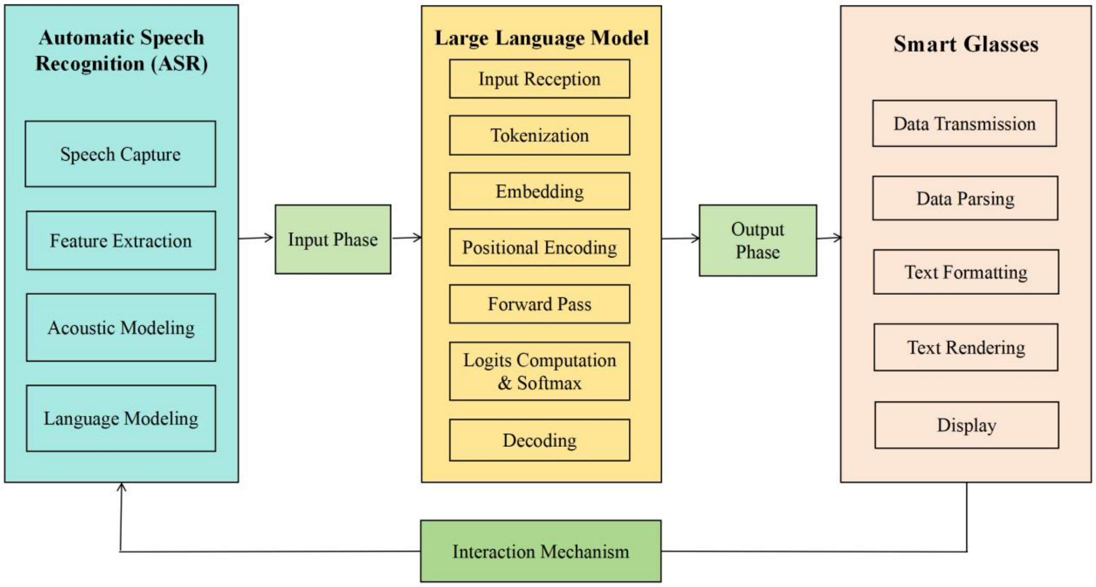


Figure 1 - ASR-LLMs-Smart Glasses Linkage Mechanism

1. Introduction

As the complex and rigorous technology in Natural Language Processing (NLP), the essence of Automatic Speech Recognition (ASR) is regarded as relying on technologies such as machine learning and artificial neural networks in many fields, which convert human spoken language into visual written Basic functions of text to facilitate the development of human-computer interaction (HCI) (Li & Hilliges, 2021;

Choutri et al., 2022; Lv et al., 2022). In the initial stage ASR has been able to respond to a limited number of human speech recognition system, and now it has developed into a clear, fluent and accurate complex response system to natural language (Padmanabhan & Johnson Premkumar, 2015).

Starting from speech capture, the operation process of ASR is closely connected with human spoken instructions as the acoustic speech signal it needs to acquire (Cooke et al., 2006). As the GPT series expands to lattice inputs and continues to change, it stimulates the fields of NLP, large language models (LLMs) and Chatbot to gradually recognize accurate and high-quality speech signals and their clear input as the key to the future human-computer interaction systems (Huang & Chen, 2019; Jeon et al., 2023). It follows an adaptive fine-tuning approach for task-specific adaptations and performs adjustments on small task-specific datasets (Hu et al, 2023; Parthasarathy et al., 2024). And for the task of enhancing NLP, LLMs has a more complex neural network and a larger training data set (Dong, 2024).

The integration of ASR technology and LLMs has aroused great interest in industry and academic research after OpenAI developed the advanced ChatGPT, which may realize the current HCI paradigm shift (Tabone & De Winter, 2023). Smart glasses are currently known portable wearable electronic computer devices that can provide enhanced visual effects and text information, which include functions such as integrated displays, cameras, microphones, and wireless connections to smartphones or other host devices (Lee & Hui, 2018). The wearer can not only see the real environment from optical display, but also see the virtual content displayed in the display, which is augmented reality (AR) concept (Kim & Choi, 2021). In recent years, companies such as OpenAI, Meta, and Google have all tried to explore designs and promote smart glasses to the global market, which provides a basis for the practicality of this research (Hou & Bergmann, 2023; Lee et al., 2023).

2. ASR-LLMs-Small Glass

Interactive cycle model is a combination of human instinct and artificial intelligence (Figure 1). It uses the linkage of ASR, large language model and smart glasses to strengthen the close relationship between human users and devices. It uses LLMs to generate text and uses ASR to identify human language information to generate what users need. Answers and solutions, and finally visualize them through smart glasses that can both see the real world and possibly see the generated text.

2.1 Automatic Speech Recognition

The speech capture stage has a crucial role in ASR that determines how accurately the speech signal can be interpreted and processed by subsequent stages, as a poor quality initial capture can lead to misunderstandings and errors in the final transcription. In the process of converting sound waves into electrical signals.

Microphone arrays are speech capture devices used to convert physical speech signals of air pressure changes into electrical signals (Seltzer, 2003). While the hands-free functionality enables the microphone to intelligently capture not only the meaning of speech desired by the user and LLMs, but also other unwanted noise-causing signals present in the user's location and environment (Wagner et al., 2024).

As shown in previous literature, the final result obtained in the speech capture stage is represented by the digital time series of the speech signal (Nasereddin & Omari, 2017). It needs to preserve the characteristics of the original speech as much as possible, and then extract the characteristics of these digital signals in subsequent stages (Shrawankar & Thakare, 2013).

Feature extraction can apply a window function to the speech signal, which is usually represented as a Hamming window or Hanning window, which divides the continuous speech signal into smaller overlapping frames, which are usually about 20-30 ms long (Ghodasara et al., 2016). When the speech signal is framed, the architecture performs a fast Fourier transform (FFT) for each frame, which converts the signal from the time domain to the frequency domain. Mel spectral coefficients (MFCC) or perceptual linear prediction (PLP) coefficients are usually extracted from frequency domain frames (Langkvist et al., 2014).

Acoustic modeling is a process step in ASR techniques to represent statistical models of the relationship between acoustic signals and speech units (Tachbelie et al., 2014). In modern ASR systems, the most commonly used feature type is the Mel-frequency cepstral coefficient (MFCC) that captures the power spectrum of an audio signal in a manner that approximates the response of the human auditory system (Jin et al., 2013).

Essentially the acoustic modeling task belongs to the pattern recognition problem (Garg & Sharma, 2016) that aims to find the sequence of speech units most likely to produce the observed acoustic features (Chang et al., 2022). This task is usually accomplished by the statistical model of Hidden Markov Model (HMM), because this model is able to process data sequences and incorporate time dependencies between different parts of the speech signal (Monir et al., 2021).

Language modeling is the last key stage of the ASR part of this research. It revolves around the construction of a statistical model that predicts the likelihood of a sequence of words to overcome accuracy challenges due to linguistic uncertainty (Wei et al., 2023).

2.2 Large Language Model

LLMs are highly advanced NLP models based on the transformer architecture. Through continuous fine-tuning of the design, it can employ a self-attention mechanism to capture context and construct meaningful text output based on correlation predictions (Chang et al., 2024). Another key element is its positional embedding layer, which assigns tokens to high-dimensional vectors according to their corresponding positions in the input sequence (Hadi et al., 2024).

In the operation process of LLMs, input reception is a crucial link, responsible for receiving text data for further processing, and is directly connected to the language modeling stage of the ASR system (Patel et al., 2022). Tokenization in LLMs is mainly achieved through a technology called byte pair encoding (BPE) (Tavabi & Lerman, 2021). At the heart of the technology lies contextual tokenization, which is processed by splitting textual data into individual units or tokens. This process is driven by BPE, which is designed to cope with infinite vocabularies in a memory-efficient manner (Petrov et al., 2023). Specifically, initial tokens are typically individual characters drawn from a base vocabulary, and longer tokens are then generated by iteratively merging the most frequently adjacent pairs of symbols in the dataset (Xu & Zhou, 2022).

The core functionality of tokenization is that it breaks the text string coming from the input receiver into smaller units. They usually correspond to words or subwords, but the specific granularity may vary depending on different application scenarios (Liu et al., 2021).

Embedding is the conversion of tokens into vectors for the model after tokenization to achieve the ability to understand and generate human language. In this process, each token in the tokenization stage is mapped into a high-dimensional vector (Rothman & Gulli, 2022). The final result is a matrix, with each row representing the d-dimensional embedding of a token in the input text, and provides the basis for subsequent positional encoding (Zheng et al., 2021; Naeve et al., 2024). However, due to the architectural limitations of the model itself, LLM cannot naturally identify sequential information in sequences (Chen et al., 2021). To solve this problem, LLM ensures that the model can grasp the relative position of markers in the sequence by adding position encoding to marker embeddings (Kazemnejad et al., 2024).

Considering that the conversion layer consists of a multi-head self-attention mechanism and a feedforward neural network, the embedded information provides the core basis for the processing of the conversion layer (Reza et al., 2022; Zheng et al., 2024). Multi-head self-attention enables the model to deeply understand the complex structure of grammatical and semantic relationships in text by measuring the relative importance of each token compared to other tokens in the context (Vaswani, 2017; Xiao et al., 2022).

When the position encoding vector is generated, the system adds it to the corresponding mark embedding vector in the embedding stage. The result of this process is a set of vector sequences, where each vector not only contains the semantic information of the tags in the embedding stage, but also incorporates the relative position information in the sequence provided by the position encoding (Naveed et al., 2023; Naeve et al., 2024). The generated vector sequence (encapsulating the semantic representation of the mark and its sequence in the sequence position) into the forward pass phase (Luo et al., 2023). Each transformer's decoder receives input from the self-attention and feed-forward neural network, executes, and so on. At this stage, tokens, embeddings, and positional encodings are passed through a stack of transformer-decoders (Raiaan et al., 2024).

Softmax function to generate a probability distribution over the vocabulary for each token in the sequence. It essentially normalizes the logits so that they sum to 1, and the size of each logit determines the probability of the corresponding label.

2.3 Smart Glasses

Smart glasses are considered as wearable devices that superimpose digital data onto the user's real-world view (Surti & Mhatre, 2021), and their functionality usually depends on three components: optical system, processing unit, and user interface (Czuszynski et al., 2015). In smart glasses, the processing unit is considered the smart brain of the device, responsible for coordinating and managing core functions (2017). The main function of this unit is to process input data from various sensors, execute specified applications, and present the processing results to the user in the form of display (RajKumar et al., 2019). The data transfer process uses a structured approach to transfer the decoded text from the LLM to the smart glasses. Through the optical system, the processed data is superimposed into the user's real-time field of view, thereby visually displaying the processing results to the user (Xu et al., 2024)

The purpose of data parsing is to correctly interpret the data received from the previous “data transfer” stage (Novac, 2022; Soldner et al., 2022). It breaks down a string of data into smaller tokens, and these components are easier to manage and interpret by the smart glasses' computing system. Parsing is usually guided by a set of syntactic rules, which in this case will be dictated by the text formatting and display requirements of the smart glasses user interface.

The concept of "Text Formatting" in the context of smart glasses involves the integration of parsed text data into a format that is both visually appealing and user-readable (Firstenberg & Salas, 2014). Text formatting operations are a key factor in ensuring user satisfaction and system usability that involves aspects such as font size, typeface, text alignment, spacing, color, and other visual attributes that can affect readability and the overall user experience (Guo et al., 2022). Text rendering facilitates the implementation of processed and structured data on the interface of smartglasses. Acquired through text formatting, depending on the display technology of the smartglasses (Lee & Hui, 2018; Huang, 2022). In addtion, the role of the display is to successfully communicate the output of the model to the user in a readable, clear and efficient manner.

3. Proposed Algorithms

In contrast, mathematical formula allows systematic variation of model parameters, which facilitates the exploration of operational boundaries and conditions under which interaction models can perform optimally (Bubeck et al., 2023).

The architecture is divided into four conceptual steps from exposure to user speech: speech capture, feature extraction, acoustic modeling, and language modeling. Examining performance requires a thorough evaluation of the different components to reflect their unique contributions to the overall functionality of the ASR system.

Word Error Rate (WER) provides an overall assessment of ASR performance by quantifying ASR systems' quantifying insertions I, deletions D, and substitutions S compared to human-transcribed reference texts (Ali & Renals, 2018) . The calculation takes into account the total number of words N in the reference text. This key evaluation indicator is expressed as:

$$\mathrm{WER} = (S + D + I) / N$$

- Voice capture: While there is no clear metric to evaluate the quality of voice capture, an indirect measure, the signal-to-noise ratio (SNR), can be used. A higher SNR indicates superior capture quality.

- Feature extraction: Feature extraction in ASR usually uses Mel-frequency cepstral coefficients (MFCC). While there is no direct measure of feature extraction quality, its impact can be inferred from WER. A high WER may indicate insufficient feature extraction.

- Acoustic Modeling: Acoustic model performance can be measured using the Frame Error Rate (FER), a metric that records the proportion of misclassified frames:

$$\text{FER = Incorrectly classified frames / Total frames}$$

- Language Modeling: Perplexity is a common metric for evaluating language models. A higher language model is indicated by a lower perplexity value. For a test set of N words *W1*, *W2*, ..., *WN*, perplexity is defined as:

$$\text{Perplexity} = (\textit{Product from i=1 to N of } (\textit{1/P(Wi}|\textit{Wi-1, ..., W1})))\hat{}^{(1/N)}$$

The combination of the above formulas provides a comprehensive framework for critically evaluating the accuracy of ASR systems.

The coherence of speech-to-text translation of an automatic speech recognition (ASR) system is intricately linked to its ability to maintain logical and contextual continuity in the output text (Narisetty et al., 2022).

Below is a breakdown of potential metrics for the four main components of an ASR system:

- Speech Capture: High quality voice capture ensures that the ASR system has a good starting point for translation. Any misunderstanding at this stage will affect subsequent stages. While there is no specific mathematical formula to measure coherence at this stage, a high signal-to-noise ratio (SNR) will enhance the clarity of captured speech and indirectly improve coherence.
- Feature extraction: In ASR, feature extraction techniques such as Mel-frequency cepstral coefficients (MFCC) are used. Sufficient feature extraction can preserve the essential characteristics of speech, which is necessary for coherence.
- Acoustic Modeling: An acoustic model converts an acoustic signal into a sequence of phonemes or words. The quality of the model will greatly affect the coherence of the output. A low frame error rate (FER) can indicate that the acoustic model is performing well, producing coherent output.

$$\text{FER} = \textit{Misclassified frames / Total frames}$$

- Language Modeling: Coherence is primarily measured at this stage. Perplexity, the inverse probability of the test set, normalized by the number of words, can measure coherence. A lower perplexity score indicates better coherence because the language model can more accurately predict subsequent words in the sequence.

$$\text{Perplexity} = (\textit{product from } i\textit{=1 to } N\ (1/P(Wi|Wi\text{-}1, ..., W1)))\text{^}^{(1/N)}$$

Given that the nature of coherence is a more qualitative than quantitative feature, the above formulations and assessments provide indirect insight into the coherence of ASR systems.

Delay rate is critical for the seamless functioning of graphs through timely speech-to-text conversion. It is quantified as the time difference between speech input and its text representation output (Wang et al., 2022). Each stage contributes to the total latency and can be rigorously evaluated using specific metrics:

Speech capture: The delay rate of speech capture is highly hardware dependent and can be affected by factors such as microphone quality, signal-to-noise ratio, and network delay rate.

Feature extraction: Feature extraction is a computationally intensive task, so delay rate can be significant, depending on the hardware used. Fetch time "T_{fe}" is often used as a measure of this delay. The exact time depends on factors such as the complexity of the extraction algorithm and the computing power of the hardware.

Acoustic Modeling: Acoustic modeling that translates speech features into a sequence of phonemes or words also contributes to overall latency. This delay "T_{am}" can be calculated by the time it takes to run the acoustic model on the extracted features.

Language Modeling: Language modeling adds a further delay "T_{lm}" as it involves predicting the probability distribution of possible words following a given history of words.

The total delay "D_{total}" can be approximated by summing the individual delays:

$$D_{total} = T_{fe} + T_{am} + T_{lm}$$

4. Experiment

Since this architecture is not a model that already exists and is practiced in the current industry and academia, it is impossible to analyze the model by obtaining data. And the model represents a complex intersection of technologies involving ASR, LLMs, and smart glasses for information display. Mathematical formulas are suitable for processing and explaining large and complex data sets in the ASR-LLMs-Smart Glasses model and obtaining objective and valid results from them. Predictability is the strength of mathematical formulations that extend beyond retrospective analysis by discerning patterns and relationships in data that include predictions of future outcomes or trends in interactive loop models.

4.1 Experimental Setup

The researcher design of this research is based on a complex recurrent interaction model that contains three key modular components: ASR, LLMs and Smart Glasses. Its purpose is to propose ideas and analyze the linkage system operation of the above components through mathematical formula, and to simulate the collaborative operation of the interactive cycle model to realize the interactive cycle mechanism. The entire process starts with the user's voice input, which is processed by the ASR system to generate preliminary text, followed by semantic enhancement and content optimization by LLM, and finally the results are presented to the user in a visual form through smart glasses. After the user views and understands the displayed content, the system generates an audio response as the beginning of a new interaction cycle.

4.2 Implement

It is obvious that speech input is first recorded and converted into digital data, which is subsequently processed to extract unique acoustic features (Prabhavalkar et al., 2017; Ong et al., 2023). When these acoustic features are passed to an acoustic model, the model generates a sequence of phonemes as output (Dupont et al., 2005). These phonemes are then further processed by a language model, ultimately producing a text string that represents the best interpretation of the user's speech content (Kłosowski, 2022).

To pass text from ASR to the LLM module, you need to follow the operation process of the generative pre-training transformer, which includes seven main steps: receiving input, tokenizing, generating embeddings, adding positional encoding, performing forward propagation, calculating logarithm, Softmax is applied, and finally the results are generated through a decoder (Borgeaud et al., 2022; Katz & Belinkov, 2023). The architecture then assigns positional encodings to preserve the order of words, and the forward pass operation propagates these embeddings through the model. After obtaining the output logits, the Softmax operation converts them into probabilities, and the decoding step finally converts these probabilities back to human-readable text (Sun et al., 2021).

Refined text data from LLMs is data-fed to smart glasses. Smart glasses perform data parsing on the output text to identify the structural components of the text (Waisberg et al., 2024). The text rendering stage converts the formatted text into a form suitable for display, which is then presented to the user in the display step.

The researcher implemented the experiment using Python 3.13 IDLE. The research builds corresponding codes for ASR, LLMs and smart glasses in this architecture respectively. The code used in this research experiment is completely open and free. It has been uploaded to the GitHub repository, and the link is:

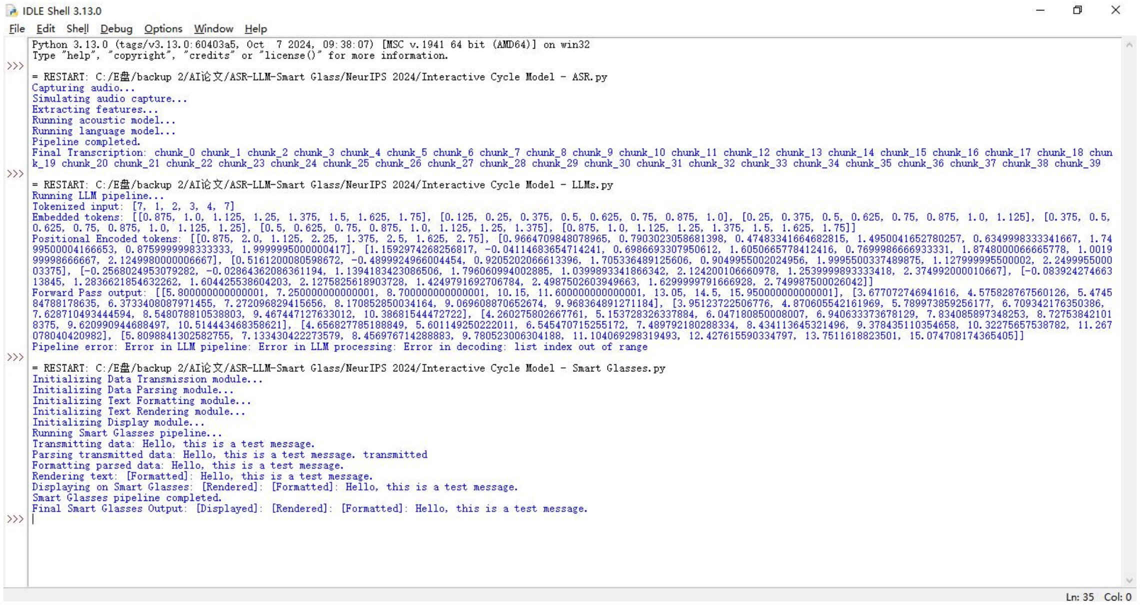
https://github.com/brucewang123456789/GeniusTrail/blob/main/Interactive%20Cycle%20Model%20-%20ASR.py

https://github.com/brucewang123456789/GeniusTrail/blob/main/Interactive%20Cycle%20Model%20-%20LLMs.py

https://github.com/brucewang123456789/GeniusTrail/blob/main/Interactive%20Cycle%20Model%20-%20Smart%20Glasses.py

5. Results

The code built based on this research completely implements the connection architecture of speech recognition, automatic language generation and smart glasses display, and can run successfully in the IDLE environment of Python 3.13. The experimental results are shown in Figure 2.



```
    Python 3.13.0 (tags/v3.13.0:60403a5, Oct  7 2024, 09:38:07) [MSC v.1941 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>> 
    = RESTART: C:/E盘/backup 2/AI论文/ASR-LLM-Smart Glass/NeurIPS 2024/Interactive Cycle Model - ASR.py
    Capturing audio...
    Simulating audio capture...
    Extracting features...
    Running acoustic model...
    Running language model...
    Pipeline completed.
    Final Transcription: chunk_0 chunk_1 chunk_2 chunk_3 chunk_4 chunk_5 chunk_6 chunk_7 chunk_8 chunk_9 chunk_10 chunk_11 chunk_12 chunk_13 chunk_14 chunk_15 chunk_16 chunk_17 chunk_18 chun
    k_19 chunk_20 chunk_21 chunk_22 chunk_23 chunk_24 chunk_25 chunk_26 chunk_27 chunk_28 chunk_29 chunk_30 chunk_31 chunk_32 chunk_33 chunk_34 chunk_35 chunk_36 chunk_37 chunk_38 chunk_39
>>> 
    = RESTART: C:/E盘/backup 2/AI论文/ASR-LLM-Smart Glass/NeurIPS 2024/Interactive Cycle Model - LLMs.py
    Running LLM pipeline...
    Tokenized input: [7, 1, 2, 3, 4, 7]
    Embedded tokens: [[0.875, 1.0, 1.125, 1.25, 1.375, 1.5, 1.625, 1.75], [0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0], [0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0, 1.125], [0.375, 0.5,
    0.625, 0.75, 0.875, 1.0, 1.125, 1.25], [0.5, 0.625, 0.75, 0.875, 1.0, 1.125, 1.25, 1.375], [0.875, 1.0, 1.125, 1.25, 1.375, 1.5, 1.625, 1.75]]
    Positional Encoded tokens: [[0.875, 2.0, 1.125, 2.25, 1.375, 2.5, 1.625, 2.75], [0.9664709848078965, 0.7903023058681398, 0.47483341664682815, 1.4950041652780257, 0.6349998333341667, 1.74
    99500004166653, 0.8759999998333333, 1.9999995000000417], [1.1592974268256817, -0.04114683654714241, 0.6986693307950612, 1.6050665778412416, 0.7699986666933331, 1.8748000066665778, 1.0019
    99998666667, 2.1249980000006667], [0.5161200080598672, -0.4899924966004454, 0.9205202066613396, 1.705336489125606, 0.9049955002024956, 1.9995500337489875, 1.127999995500002, 2.2499955000
    03375], [-0.2568024953079282, -0.02864362086361194, 1.1394183423086506, 1.796060994002885, 1.0399893341866342, 2.124200106660978, 1.2539999893333418, 2.374992000010667], [-0.083924274663
    13845, 1.2836621854632262, 1.604425538604203, 2.1275825618903728, 1.4249791692706784, 2.4987502603949663, 1.6299999791666928, 2.7499875000026042]]
    Forward Pass output: [[5.800000000000001, 7.250000000000001, 8.700000000000001, 10.15, 11.600000000000001, 13.05, 14.5, 15.950000000000001], [3.677072746941616, 4.575828767560126, 5.4745
    84788178635, 6.3733408087971455, 7.272096829415656, 8.170852850034164, 9.069608870652674, 9.968364891271184], [3.95123722506776, 4.870605542161969, 5.789973859256177, 6.709342176350386,
    7.628710493444594, 8.548078810538803, 9.467447127633012, 10.38681544472722], [4.260275802667761, 5.153728326337884, 6.047180850008007, 6.940633373678129, 7.834085897348253, 8.72753842101
    8375, 9.620990944688497, 10.514443468358621], [4.656827785188849, 5.601149250222011, 6.545470715255172, 7.489792180288334, 8.434113645321496, 9.378435110354658, 10.32275657538782, 11.267
    078040420982], [5.8098841302582755, 7.133430422273579, 8.456976714288883, 9.780523006304188, 11.104069298319493, 12.427615590334797, 13.7511618823501, 15.074708174365405]]
    Pipeline error: Error in LLM pipeline: Error in LLM processing: Error in decoding: list index out of range
>>> 
    = RESTART: C:/E盘/backup 2/AI论文/ASR-LLM-Smart Glass/NeurIPS 2024/Interactive Cycle Model - Smart Glasses.py
    Initializing Data Transmission module...
    Initializing Data Parsing module...
    Initializing Text Formatting module...
    Initializing Text Rendering module...
    Initializing Display module...
    Running Smart Glasses pipeline...
    Transmitting data: Hello, this is a test message.
    Parsing transmitted data: Hello, this is a test message. transmitted
    Formatting parsed data: Hello, this is a test message.
    Rendering text: [Formatted]: Hello, this is a test message.
    Displaying on Smart Glasses: [Rendered]: [Formatted]: Hello, this is a test message.
    Smart Glasses pipeline completed.
    Final Smart Glasses Output: [Displayed]: [Rendered]: [Formatted]: Hello, this is a test message.
>>> |
```

Figure 2 - Experimental results run through Python 3.13 IDLE

Observing the experimental results, the ASR part successfully simulated audio capture and converted it into text fragments, showing a stable modular design. However, the LLMs part had a "list index out of range" error during the decoding stage, which indicates that sequence length or index out-of-range issues may not have been properly handled when performing decoding. This error usually results from the input sequence not corresponding to the weight matrix or from inconsistencies in the processing steps. In addition, the data transmission, parsing, formatting, rendering and display modules of the smart glasses completely ran successfully, and the output results were in line with expectations, demonstrating complete message flow processing capabilities. Discussing these results, it is found that the overall architecture has potential modularity advantages, however, the LLMs part requires further optimization of the decoding logic, such as checking sequence boundaries or ensuring that input vectors match model weights. In addition, the successful execution of the smart glasses module shows that the data processing flow is relatively stable in the back-end part.

6. Limitation

After the acquired human speech information is converted into text, the interaction loop model needs to make integration and synchronization of ASR with LLMs. After ASR converts the user's speech into text, the output needs to be processed by llm to achieve natural language understanding and response generation. This means that the linking process between ASR and LLM technology and systems is infinitely complex and challenging. However, in current practice, real-time synchronization and data exchange between ASR and large language model systems, error handling, and management of potential delays or waiting times may encounter technical obstacles. And the response speed of the model may be slower than the real-time changes of the text generated by LLMs, which may cause high latency to affect user experience.

7. Summary

This research proposes an interactive cycle model centered on the "ASR-LLMs-Smart Glasses" linkage model. It is committed to promoting human-computer interaction and enhancing human capabilities through the combination of voice and vision with the model after the user wears it. The model integrates automatic speech recognition, large language model and a novel human-computer interaction architecture for smart glasses.

In order to verify the objective feasibility of the model in the absence of data due to the absence of relevant physical products, this research introduces the concept of mathematical formula to evaluate and quantify the performance of the model, which includes coherence (C), accuracy (A), Error Rate (E), Delay Rate (D) and Efficiency (Ef) metrics.

This research establishes a solid foundation for the theoretical and practical application of the ASR-LLMs-Smart Glasses model. The incorporation of this cutting-edge technology in a seamless interaction model represents a major advance in the field of human-computer interaction.

References

Ali, A., & Renals, S. (2018). Word error rate estimation for speech recognition: e-WER. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers) (pp. 20-24).

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., ... & Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. *In International conference on machine learning* (pp. 2206-2240). PMLR.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with LLMs. *arXiv preprint arXiv:2303.12712*.

Chang, S., Wang, X., Fang, T., & Qian, L. (2022). Design and Implementation of Wake-on-Voice and Command Recognition Algorithm. *In 2022 14th International Conference on Intelligent Human-Machine Systems and Cybernetics* (IHMSC) (pp. 1-4). IEEE.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... & Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology, 15*(3), 1-45.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., ... & Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems, 34*, 15084-15097.

Choutri, K., Lagha, M., Meshoul, S., Batouche, M., Kacel, Y., & Mebarkia, N. (2022). A Multi-Lingual Speech Recognition-Based Framework to Human-Drone Interaction. *Electronics, 11*(12), 1829.

Cooke, M., Barker, J., Cunningham, S., & Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America, 120*(5), 2421-2424.

Czuszynski, K., Ruminski, J., Kocejko, T., & Wtorek, J. (2015). Septic safe interactions with smart glasses in health care. *In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (EMBC) (pp. 1604-1607). IEEE.

Dong, J. (2024). Natural Language Processing Pretraining Language Model for Computer Intelligent Recognition Technology. A*CM Transactions on Asian and Low-Resource Language Information Processing, 23*(8), 1-12.

Dupont, S., Ris, C., Deroo, O., & Poitoux, S. (2005). Feature extraction and acoustic modeling: an approach for improved generalization across languages and accents. *In IEEE Workshop on Automatic Speech Recognition and Understanding* (pp. 29-34). IEEE.

Firstenberg, A., & Salas, J. (2014). *Designing and developing for Google Glass: Thinking differently for a new platform.* " O'Reilly Media, Inc.".

Garg, A., & Sharma, P. (2016). Survey on acoustic modeling and feature extraction for speech recognition. *In 2016 3rd International Conference on Computing for Sustainable Global Development* (INDIACom) (pp. 2291-2295). IEEE.

Ghodasara, V., Waldekar, S., Paul, D., & Saha, G. (2016). Acoustic scene classification using block-based MFCC features. *Detection and classification of acoustic scenes and events*.

Guo, L., Daly, E. M., Alkan, O., Mattetti, M., Cornec, O., & Knijnenburg, B. (2022). Building trust in interactive machine learning via user contributed interpretable rules. *In 27th International Conference on Intelligent User Interfaces* (pp. 537-548).

Hadi, M. U., Al Tashi, Q., Shah, A., Qureshi, R., Muneer, A., Irfan, M., ... & Shah, M. (2024). Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*.

Hu, Z., Lan, Y., Wang, L., Xu, W., Lim, E. P., Lee, R. K. W., ... & Poria, S. (2023). LLMs-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. *arXiv preprint arXiv:2304.01933*.

Huang, C. W., & Chen, Y. N. (2019). Adapting pretrained transformer to lattices for spoken language understanding. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 845-852). IEEE.

Huang, Y. (2022). AR Shopping List: Exploring the Design Space of Smart Glasses to Allow Real-time Recording with Multiple Input Formats.

Jeon, J., Lee, S., & Choi, S. (2023). A systematic review of research on speech-recognition chatbots for language learning: Implications for future directions in the era of large language models. Interactive Learning Environments, 1-19.

Jin, J. S., Xu, C., Xu, M., & Gonzalez, R. (2013). Better than MFCC audio classification features. *In The Era of Interactive Media* (pp. 291-301). Springer New York.

Katz, S., & Belinkov, Y. (2023). Interpreting Transformer's Attention Dynamic Memory and Visualizing the Semantic Information Flow of large language model. arXiv preprint arXiv:2305.13417.

Kazemnejad, A., Padhi, I., Natesan Ramamurthy, K., Das, P., & Reddy, S. (2024). The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems, 36*.

Kim, D., & Choi, Y. (2021). Applications of smart glasses in applied sciences: A systematic review. *Applied Sciences, 11*(11), 4956.

Kłosowski, P. (2022). A rule-based grapheme-to-phoneme conversion system. Applied Sciences, 12(5), 2758.

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. Pattern recognition letters, 42, 11-24.

Lee, H., Hsia, C. C., Tsoy, A., Choi, S., Hou, H., & Ni, S. (2023). VisionARy: Exploratory research on Contextual Language Learning using AR glasses with ChatGPT. *In Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter* (pp. 1-6).

Lee, L. H., & Hui, P. (2018). Interaction methods for smart glasses: A survey. IEEE access, 6, 28712-28732.

Li, Y., & Hilliges, O. (Eds.). (2021). A*rtificial intelligence for human computer interaction: a modern approach* (pp. 463-493). Cham: Springer.

Liu, X., Yang, B., Liu, D., Zhang, H., Luo, W., Zhang, M., ... & Su, J. (2021). Bridging subword gaps in pretrain-finetune paradigm for natural language generation. arXiv preprint arXiv:2106.06125.

Luo, Q., Zeng, W., Chen, M., Peng, G., Yuan, X., & Yin, Q. (2023). Self-Attention and Transformers: Driving the Evolution of Large Language Models. *In 2023 IEEE 6th International Conference on Electronic Information and Communication Technology* (ICEICT) (pp. 401-405). IEEE.

Lv, Z., Poiesi, F., Dong, Q., Lloret, J., & Song, H. (2022). Deep Learning for Intelligent Human–Computer Interaction. Applied Sciences, 12(22), 11457.

Monir, N. E., Berbara, Z., Sheikh, I., & Sahidullah, M. (2021). Self-Supervised Learning for Automatic Speech Recognition.

Naeve, Z., Mitchell, L., Reed, C., Campbell, P., Morgan, T., & Rogers, V. (2024). Introducing dynamic token embedding sampling of large language models for improved inference accuracy. *Authorea Preprints*.

Narisetty, C., Tsunoo, E., Chang, X., Kashiwagi, Y., Hentschel, M., & Watanabe, S. (2022). Joint speech recognition and audio captioning. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7892-7896). IEEE.

Nasereddin, H. H., & Omari, A. A. R. (2017). Classification techniques for automatic speech recognition (ASR) algorithms used with real time speech translation. In 2017 Computing Conference (pp. 200-207). IEEE.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... & Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Padmanabhan, J., & Johnson Premkumar, M. J. (2015). Machine learning in automatic speech recognition: A survey. IETE Technical Review, 32(4), 240-251.

Parthasarathy, V. B., Zafar, A., Khan, A., & Shahid, A. (2024). The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*.

Patel, J., Tejani, R., & Talati, B. (2022). Evaluation Of Performance Of Artificial Intelligence System During Voice Recognition In Social Conversations using NLP. In 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-4). IEEE.

Petrov, A., La Malfa, E., Torr, P. H., & Bibi, A. (2023). Language Model Tokenizers Introduce Unfairness Between Languages. arXiv preprint arXiv:2305.15425.

Prabhavalkar, R., Rao, K., Sainath, T. N., Li, B., Johnson, L., & Jaitly, N. (2017). A Comparison of sequence-to-sequence models for speech recognition. In Interspeech (pp. 939-943).

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., ... & Azam, S. (2024). A review on large Language Models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*.

RajKumar, A., Arora, C., Katz, B., & Kapila, V. (2019). Wearable smart glasses for assessment of eye-contact behavior in children with autism. In Frontiers in Biomedical Devices (Vol. 41037, p. V001T09A006). American Society of Mechanical Engineers..

Reza, S., Ferreira, M. C., Machado, J. J. M., & Tavares, J. M. R. (2022). A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. Expert Systems with Applications, 202, 117275.

Rothman, D., & Gulli, A. (2022). Transformers for Natural Language Processing: Build, train, and fine-tune deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, and large language model-3. Packt Publishing Ltd.

Seltzer, M. L. (2003). Microphone array processing for robust speech recognition (Doctoral dissertation, Carnegie Mellon University).

Shrawankar, U., & Thakare, V. M. (2013). Techniques for feature extraction in speech recognition system: A comparative study. arXiv preprint arXiv:1305.1145.

Söldner, R., Rheinländer, S., Meyer, T., Olszowy, M., & Austerjost, J. (2022). Human−Device Interaction in the Life Science Laboratory. In Smart Biolabs of the Future (pp. 83-113). Cham: Springer International Publishing.

Sun, S., Zhang, Z., Huang, B., Lei, P., Su, J., Pan, S., & Cao, J. (2021). Sparse-softmax: A Simpler and Faster Alternative Softmax Transformation. arXiv preprint arXiv:2112.12433.

Surti, P., & Mhatre, P. (2021). Smart Glasses Technology.

Syberfeldt, A., Danielsson, O., & Gustavsson, P. (2017). Augmented reality smart glasses in the smart factory: Product evaluation guidelines and review of available products. Ieee Access, 5, 9118-9130.

Tabone, W., & De Winter, J. (2023). Using ChatGPT for human−computer interaction research: a primer. *Royal Society Open Science, 10*(9), 231053.

Tachbelie, M. Y., Abate, S. T., & Besacier, L. (2014). Using different acoustic, lexical and language modeling units for ASR of an under-resourced language−Amharic. Speech Communication, 56, 181-194.

Tavabi, N., & Lerman, K. (2021). Pattern Discovery in Time Series with Byte Pair Encoding. arXiv preprint arXiv:2106.00614.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wagner, D., Churchill, A., Sigtia, S., Georgiou, P., Mirsamadi, M., Mishra, A., & Marchi, E. (2024). A Multimodal Approach to Device-Directed Speech Detection with Large Language Models. *In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP) (pp. 10451-10455). IEEE.

Waisberg, E., Ong, J., Masalkhi, M., Zaman, N., Sarker, P., Lee, A. G., & Tavakkoli, A. (2024). Meta smart glasses—large language models and the future for assistive glasses for individuals with vision impairments. *Eye, 38*(6), 1036-1038.

Wang, W., Ren, S., Qian, Y., Liu, S., Shi, Y., Qian, Y., & Zeng, M. (2022). Optimizing alignment of speech and language latent spaces for end-to-end speech recognition and understanding. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7802-7806). IEEE.

Wei, C., Wang, Y. C., Wang, B., & Kuo, C. C. J. (2023). An overview on language models: Recent developments and outlook. arXiv preprint arXiv:2303.05759.

Weng, Z., Qin, Z., Tao, X., Pan, C., Liu, G., & Li, G. Y. (2023). Deep learning enabled semantic communications with speech recognition and synthesis. IEEE Transactions on Wireless Communications.

Xiao, L., Hu, X., Chen, Y., Xue, Y., Chen, B., Gu, D., & Tang, B. (2022). Multi-head self-attention based gated graph convolutional networks for aspect-based sentiment classification. Multimedia Tools and Applications, 1-20.

Xu, T., & Zhou, P. (2022). Feature Extraction for Payload Classification: A Byte Pair Encoding Algorithm. In 2022 IEEE 8th International Conference on Computer and Communications (ICCC) (pp. 1-5). IEEE.

Xu, Z., Xu, H., Lu, Z., Zhao, Y., Zhu, R., Wang, Y., ... & Shang, L. (2024). Can Large Language Models Be Good Companions? An LLM-Based Eyewear System with Conversational Common Ground. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 8*(2), 1-41.

Zheng, J., Ramasinghe, S., & Lucey, S. (2021). Rethinking positional encoding. arXiv preprint arXiv:2107.02561.

Zheng, Z., Wang, Y., Huang, Y., Song, S., Yang, M., Tang, B., ... & Li, Z. (2024). Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*.