Data and reports of the speech recognition system with memory function based on the server side

(Non-scientific report)

Development time: July 18, 2023 to July 21, 2023

Publication status: csdn, git

Researcher: Wu Xiang

Research unit: New Shijia Intellectual Property Institute, Shanghai Yuzhou-Lingxi AI Laboratory

Patent registration number or national application Number:

Software copyright registration number or application number:

Identity license: 6060933

Project registration No.: IDN _ 2023071801091

Statement to protect project research reports and data intellectual property rights

Dear recipients,

I, Wu Xiang, as a researcher of [the research data and report of the server-based speech recognition system with memory function], hereby send you a statement about the research report of this project. This research report is the result of a project supported and sponsored by the New Family Intellectual Property Institute (hereinafter referred to as "New Family"), and we will ensure the legitimacy of its intellectual property and protection.

The purpose of this statement is to emphasize the protection of intellectual property rights for project research reports and prevent them from being misappropriated, abused or unauthorized use by others after disclosure. The research report contains a wealth of research results and knowledge, through the hard work of the team and multiple cooperation collection. In order to safeguard the legitimate rights and interests of intellectual property rights and protect the uniqueness of the project results, we solemnly affirm the following matters:

Intellectual property rights: The intellectual property rights in this study belong to a new family or to a specific partner if otherwise clearly agreed in the relevant agreements and contracts. Without the explicit authorization of New Family or other relevant partners, no individual or organization shall claim the ownership of the research report.

Confidentiality and Authorization: As a project investigator, I will always do my best to maintain the confidentiality of this study report. I will not disclose or provide any part or all of this study report to any unauthorized third party unless explicitly authorized in writing.

Prohibition of misappropriation and abuse: Any unauthorized use, reproduction, modification, dissemination or resale of this study report is regarded as an infringement of the new intellectual property rights of the family. For the violations of intellectual property rights, we will pursue the corresponding legal responsibilities according to the law.

Legal use: If an individual or organization wishes to legally use the contents of this study, it must obtain explicit written authorization from the new family and comply with the relevant use conditions and restrictions.

Sharing and quoting: We welcome academics and related fields to cite the content of this research report, but we must comply with the academic norms and indicate the source, indicating the source of the research report and the intellectual property rights of the new family.

We have a firm belief in the protection of intellectual property rights, and we also thank all parties for their attention and support to the project. It is hoped that this statement will increase the society's awareness of intellectual property protection and jointly maintain the healthy development of knowledge innovation and academic research.

In my statement,

Wu Xiang
Study data and report date of the server-based speech recognition system: July 21,2023

Project registration No.: IDN _ 2023071801091

abstract

The research objective of this project is to explore whether the speech recognition system can conduct memory cache processing on the server side and propose a better processing method. The project adopted python, developed for the base script, at the same time with the local computer as the hypothesis, write a common speech recognition system and another called segment processing speech recognition system code, and use large-scale data simulation technology for analysis, obtained the two different cpu usage, finally under the comparison named segment processing speech recognition system code under the same conditions and ordinary speech recognition system, can better memory cache and processing on the server.

Project registration No.: IDN _ 2023071801091

Enter the wav algorithm

Import the necessary libraries: The code first imports the modules and libraries that need to be used, including the os, pyaudio, and wave modules.

Set recording parameters: at the beginning of the code, define some recording parameters, such as audio block size, sampling number, number of sound channel, sampling rate, recording duration, etc. You can adjust these parameters as needed.

Create an PyAudio object: via the pyaudio. PyAudio () Create PyAudio objects for audio stream processing.

Turn on the microphone for recording: use the audio. The open () opens the microphone for recording, and sets the corresponding parameters, such as sampling format, sound channel number, sampling rate, etc.

Recording process: Read audio data from the microphone and add each

audio block to the frames list. Stop the recording flow: After the

recording ends, stop the recording flow, close the microphone, and

terminate the PyAudio object.

Merge audio data: merge the audio blocks in the frames list into complete audio data.

Create the output and backup directory: check and create the output and backup directory (if they do not exist).

Process duplicate files: If named "record" already exists in the output directory. Ffile for wav ", which the code will renaming to" record _1.wav "," record _2.wav ", etc. in the backup directory.

Save as a WAV file: save the audio data as a WAV file and set the corresponding number of sampling channels, sampling width, and frame rate.

Exception handling: Using try-except, block capture keyboard interrupt (press Ctrl + C) to stop the recording and turn off the stream and PyAudio objects correctly.

matters need attention:

Before running the code, make sure that the pyaudio library is installed and that the microphone is available.

Project registration No.: IDN _ 2023071801091

The Wav transformation algorithm

1. Import the necessary libraries: The code first imports the modules and libraries to be used, including os, shuti l and speech _recognition modules.

2. Create a speech recognizer object: use the ` sr. Recognizer ()` Create a speech recognizer object r for subsequent audio recognition.

3. Set the directory path: define the path for the input directory (INPUT _ DIR), output directory (OUTPUT _ DIR), and backup directory (BACKUP _ DIR).

4. recognize _audio _file function: This function is used for speech recognition of individual audio files. It uses the ` sr. AudioFile ()` Open the audio file and then pass the ` r.record ()` Load the audio data. Then proceed with Google's voice recognition service (` r.recognize _google ()`) Convert the audio to text. If the identification is successful, the identified text is returned, otherwise print the relevant error message and return None.

5. Move _ existing _ files function: This function is used to move the existing output files to the backup directory. If a file with the same name already exists in the output directory, the function renames it and moves it to the backup directory.

6. Get the list of files in the input directory: using ` os.l istdir ()` Get all files in the input directory and trathrough them.

7. For to. The file ending with wav, performs speech recognition: for each WAV audio file, call the `recognize _audio _file ()` function for speech recognition to obtain the recognized text.

8. If the text is successfully identified, save it as a text file: if the recognition is successful, the identified text is saved to the output directory, using the same file name as the input audio file, but replaced with. The txt file suffix.

9. If the identification fails, the corresponding output file will not be created, and the corresponding error message will be printed.

10. Notes:
-Before running the code, make sure that you have the speech _recognit ion library installed and an audio file in WAV format under the input directory (INPUT _ DIR).
-The voice recognition service in the code uses Google's service and therefore requires a connection to the Internet. If you have no access to Google services in the domestic network environment, you can try to use other voice

recognition services, such as iFlytek, and adjust the code to accommodate the service accordingly.
The- `generate _unique _filename ()` function is used to generate a unique file name to avoid heavy name files in the output directory.

Project registration No.: IDN _ 2023071801091

Sectional processing

1. Define the path of two script files: `script 1_path ` and `script 2_path ` point to two Python script files respectively.

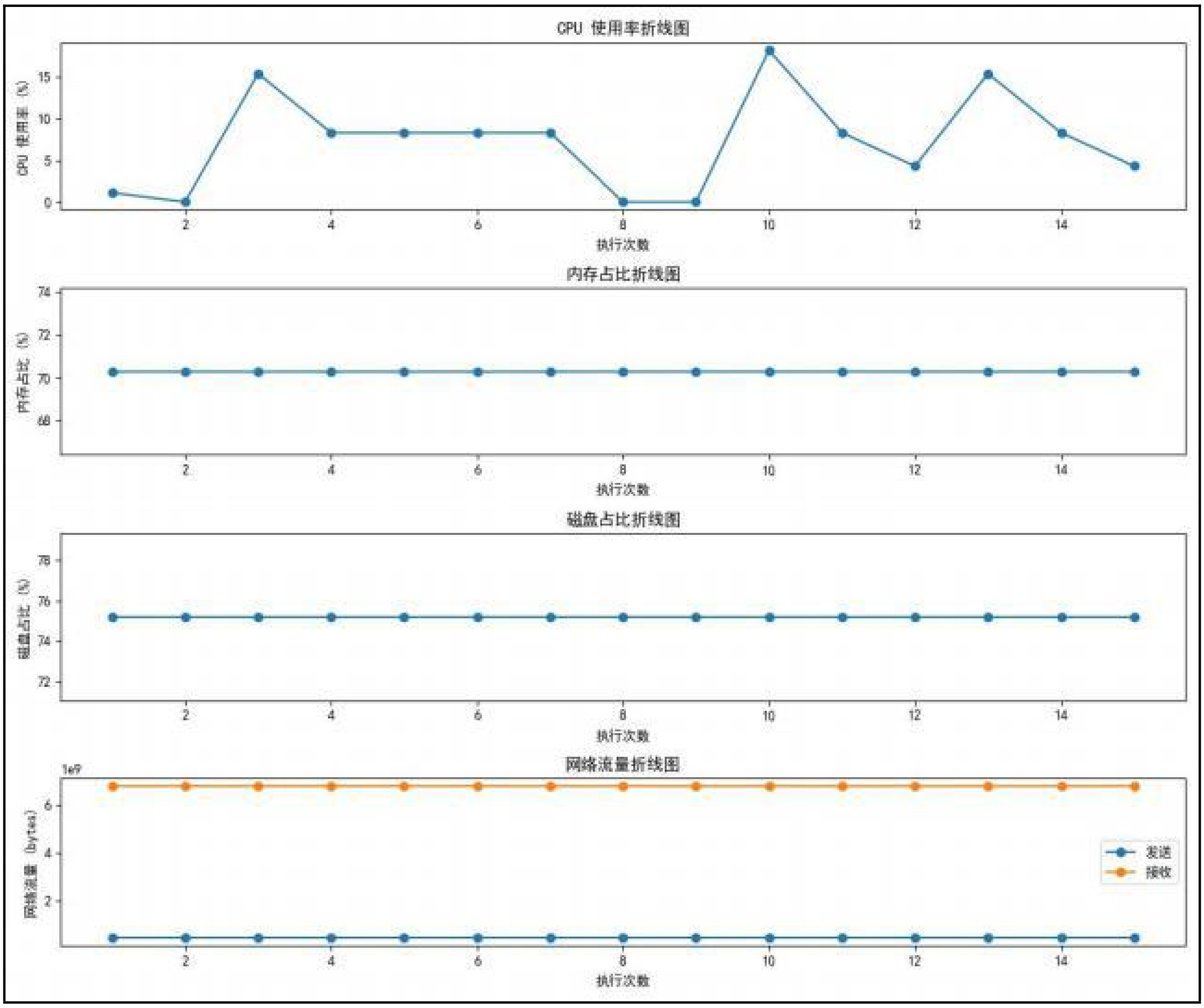
2. Execute the first script: " 2023_ latest entry. The py ": using the `subprocess. The run ()` runs the `python ` command to start the Python interpreter and pass the `script 1_path ` as an argument. This performs the first script text
pi。

3. Execute the second script: "2023" _ the latest output.py ": After the first script is executed, the code continues to use `subprocess again. The run ()` runs the `python ` command to start the Python interpreter and pass the `script 2_path ` as an argument. This performs a second script file.

matters need attention:
-Before running the code, make sure that you have correctly specified the paths to both script files (`script 1_path ` and `script 2_path `).
-Ensure that Python is already installed in your system and can be run on the command line.
-Ensure that both script files exist and contain a valid Python code. Before running, it is recommended that you run both script files manually to ensure that they work properly when executed independently.
- `subprocess . The run ()` function waits for the child process to execute before executing subsequent code, so there may be a waiting time when running the script, depending on how long the script runs. If you want to execute two scripts in parallel, please refer to `subprocess. The Popen ()` function to implement the parallel execution

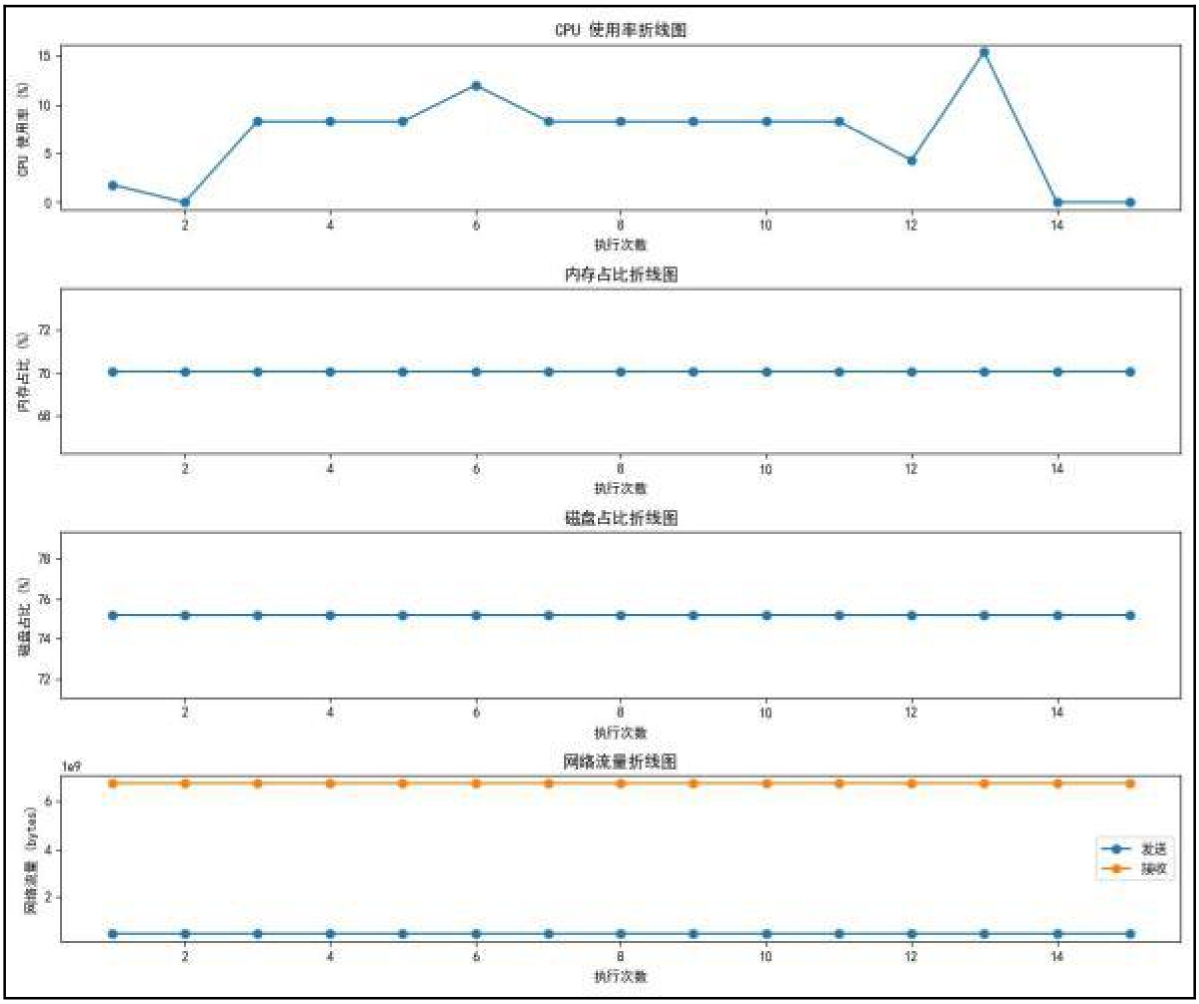
Project registration No.: IDN _ 2023071801091

Data comparison: _ ordinary speech recognition



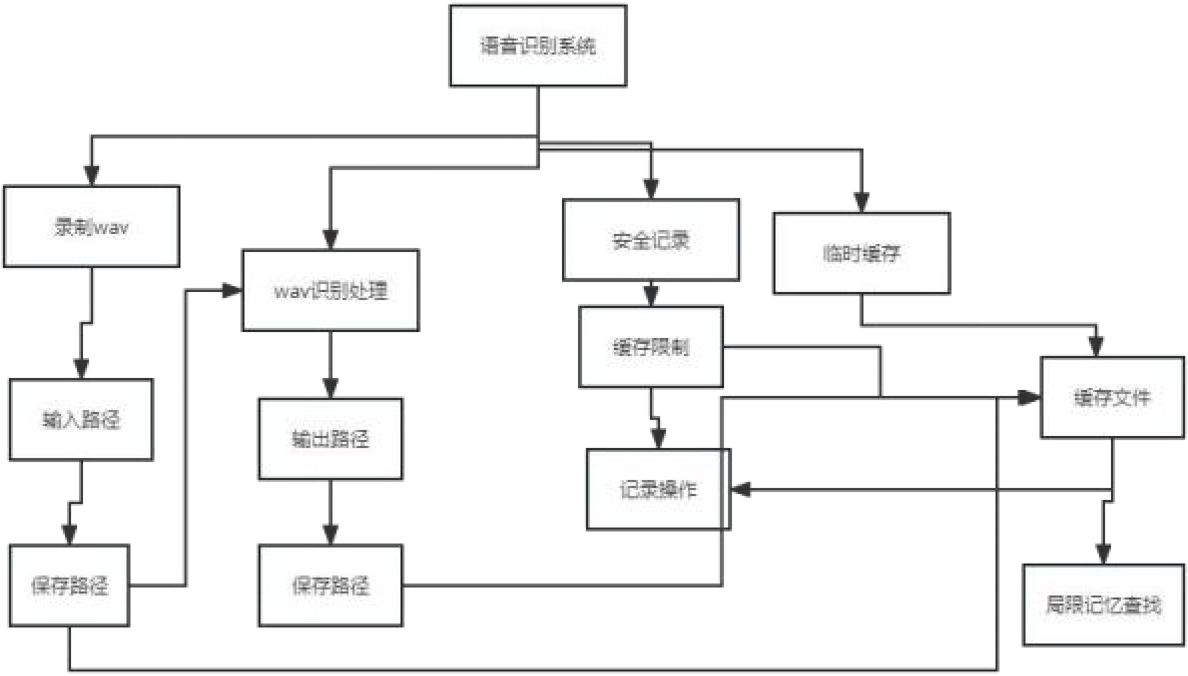
Project registration No.: IDN _ 2023071801091

The segmentation processing of speech recognition



Project registration No.: IDN _ 2023071801091

fundamental algorithm



Project registration No.: IDN _
2023071801091

Memory cache

The project here assumes that the local computer is a cloud server, which has set up four different documents

spin	2023/7/21 8:36	文件夹
spin_1	2023/7/21 8:36	文件夹
word	2023/7/21 8:03	文件夹
word_1	2023/7/21 8:03	文件夹

Here we use the method of _ + 1 to increase the cache folder to make it memory storage.

(1) Spin >>>spin _1

Spin is the document for recording audio, and spin _1 is the server cache storage

(2) word>> word _1

iabove

* The purpose of establishing a memory buffer folder is to enable it to retrieve the memory and conduct secondary training when the content is required.