УТВЕРЖДЕН

ДССЛ.00102 - ЛУ

СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«3i Speech Diarization SDK»

Описание применения

ДССЛ.00102

2015

АННОТАЦИЯ

Настоящий документ предназначен для ознакомления со специальным программным обеспечением (СПО) «3i Speech Diarization SDK» и содержит описание интерфейса программирования (API) для программистов, обеспечивающих использование 3i Speech Diarization SDK в качестве модуля, встраиваемого в другое программное обеспечение.

В разделе 1 приводятся сведения о назначении и составе 3i Speech Diarization SDK

В разделе 2 указаны требования к программно-техническим средствам, необходимым для работы 3i Speech Diarization SDK

В разделе 3 указывается описание задач, решаемых 3i Speech Diarization SDK, даются сведения об используемых технологиях.

В разделе 4 даются сведения об установке 3i Speech Diarization SDK

В разделе 5 приводится описание интерфейса программирования 3i Speech Diarization SDK

В разделе 6 даются сведения о входных и выходных данных 3i Speech Diarization SDK

В разделе 7 приводятся основные сообщения оператору при работе с 3i Speech Diarization SDK

По всем вопросам, связанным с использованием СПО «3i Speech Diarization SDK» можно обращаться по электронной почте support@dss-lab.ru или по телефону +7 (495) 645-44-70 по будним дням с 10 до 18 часов, время московское.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Специальное программное обеспечение «3i Speech Diarization SDK» предназначено для решения задач распознавания речи, диаризации речевых диалогов (разметки диалога по дикторам). 3i Speech Diarization SDK используется в качестве библиотеки модулей, встраиваемых в другое программное обеспечение, предоставляя разработчику соответствующий функционал API.

API (сокр. англ. Application Programming Interface, интерфейс программирования приложений, интерфейс прикладного программирования) – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах.

API модулей 3i Speech Diarization SDK относится к классу библиотек с динамическим связыванием (dynamic link libraries, DLL).

К основным преимуществам приложения, предоставляющего DLL API в качестве инструмента доступа к функциональным возможностям, можно отнести:

- надёжность (за счет отсутствия необходимости взаимодействия с удаленным модулем);
- производительность (за счет использования кэша и подгрузки кода и данных в пространство адресов приложения);
- прозрачность системы взаимодействия;
- легкость внесения изменений;
- масштабируемость.

2. УСЛОВИЯ ПРИМЕНЕНИЯ

2.1. Для функционирования 3i Speech Diarization SDK необходима вычислительная система с параметрами не хуже:

- CPU Intel Core i7 – 5820K 3.3 ГГц (6 физических вычислительных ядер);
- ОЗУ 16 ГБ;
- 100 Гб свободного места на жёстком диске.

2.2. Для функционирования 3i Speech Diarization SDK на вычислительной системе должно быть установлено следующее общее программное обеспечение:

- операционная система – Microsoft Windows 7 SP1 или выше;
- распространяемый пакет Microsoft VC 2013 Redist (x64).

3. ОПИСАНИЕ ЗАДАЧИ

1.1. Технологии работы с речевыми данными

Технология разделения речевого сигнала реализована на многослойной нейронной сети (DNN – от сокр. Deep Neural Network), обученной извлекать из краткосрочной спектральной характеристики речевого сигнала признаки, характеризующие голос диктора. Каждый такой вектор признаков называется «глубоким» вектором или d-вектором. Расстояние между двумя такими векторами будет малым, если они принадлежат одному диктору, и большим, если разным. Это свойство позволяет обнаружить точки смены говорящего, а также «объединить» фрагменты, в которых присутствует голос одного диктора.

1.2. Состав 3i Speech Diarization SDK

− 3i Speech Diarization SDK – модуль предназначен для диаризации/сегментации дикторов.

3. ВЫЗОВ И ЗАГРУЗКА ПРОГРАММЫ

1.1. Установка 3i Speech Diarization SDK

Установка модулей (библиотек) 3i Speech Diarization SDK осуществляется переносом (копированием) архивов с модулями с загрузочного диска в требуемую директорию на жестком диске сервера. После переноса (копирования) архива модуля динамические библиотеки (DLL) модуля 3i Speech Diarization SDK разархивируются из него в требуемую директорию на жестком диске сервера.

1.2. Получение лицензионного файла-ключа

Бинарные файлы модулей 3i Speech Diarization SDK защищены от нелицензионного использования и копирования, лицензионное использование предполагает получение файла-ключа, который специфичен для системы пользователя и бинарного файла модуля. Таким образом, полученный файл-ключ нельзя использовать на другой машине или для другого модуля, что исключает нелицензионное копирование и использование модулей. Файл-ключ может быть получен у технической поддержки разработчика в случае наличия у пользователя лицензии. Процедура получения файла-ключа такова:

1) В папке каждого модуля находится директория GenHardID, содержащая исполняемый файл GenHardID_console.exe, генерирующая уникальный идентификационный код системы пользователя. После распаковки архива модуля следует запустить файл.
2) Результатом работы GenHardID_console.exe является файл hardware_id.hid, содержащий идентификационный код системы пользователя. Это код одинаков для любого модуля и характеризует систему (машину) пользователя, таким образом, допустимо получить этот файл один раз. Файл hardware_id.hid появляется в той же папке, в которой находится исполняемый файл GenHardID_console.exe.
3) Файл идентификационного кода системы hardware_id.hid следует передать в техническую поддержку 3i, сопровождая информацией о лицензии.
4) При подтверждении пользовательской лицензии, техническая поддержка высылает файл-ключ пользователю. Файл-ключ уникален для каждого исполняемого файла (библиотеки, DLL). Его следует положить рядом с тем бинарным файлом (исполняемым модулем, DLL), для которого файл-ключ был сгенерирован. После этого, модуль будет способен запуститься и работать.

4. ВЫПОЛНЕНИЕ ПРОГРАММЫ

1.1. Описание функционала модуля API

Модули 3i Speech Diarization SDK имеют классический интерфейс, характерный для динамически подгружаемых библиотек (dll). В комплекте с каждым модулем поставляется:

- бинарный файл динамически связываемой библиотеки (с расширением dll);
- бинарный файл прокси-библиотеки для статического связывания приложения пользователя с динамически связываемой библиотекой (с расширением lib);
- файл-хидер, содержащий исходный код интерфейса модуля на c++ для сборки связывания динамической библиотеки модуля и приложения пользователя (с расширением h);

API каждого модуля описано далее.

1.1.1. Модуль 3i Speech Diarization SDK

Инициализация библиотеки

Прежде, чем приступить к работе с библиотекой, необходимо её инициализировать – загрузить конфигурационный файл, содержащий настройки модуля и пути к моделям UBM.

API функции:

VST_API(int) init(char const* pathToConfig);

pathToConfig путь к конфигурационному файлу (*.ini)

Функция возвращает код согласно таблице возвращаемых кодов.

Деинициализация библиотеки

По окончании работы с библиотекой необходимо выгрузить данные, которые были загружены при инициализации библиотеки.

API функции:

VST_API(void) deInit();

После вызова данной функции будет заблокирована возможность работы с некоторыми функциями библиотеки, а именно processDialog_FromFile () и processDialog_FromBuf (). Функция deInit() будет «ждать» окончания работы всех запущенных ранее функций из числа перечисленных, а по завершении их работы выполнит выгрузку данных. Любая из этих функций, вызванная в момент «ожидания», завершится кодом IS_BLOCKED=8.

Функция возвращает код согласно таблице возвращаемых кодов.

Сегментация речевого сигнала из буфера в памяти

Сегментирует аудиосигнал, находящийся в виде буфера в памяти, на участки с метками, отмечающими речь какого из двух дикторов находится в данном участке.

API функции:

VST_API(vst_res_ptr) processDialog_FromBuf(int & resCode, short const* audioSamples, int samplesAmount);

resCode адрес для записи завершения работы функции

audioSamples указатель на массив отсчетов

samplesAmount количество отсчетов в аудио буфере

Функция возвращает указатель на массив блоков, содержащих структуры VST_results

```
typedef struct vst_res_t
{
int SegmentsAmount;
int SpeakersAmount;
int SignalDuration; // ms
vst_res_segment_ptr Segments; // set of VST-segments
} vst_res_t;
```

Сегментация речевого сигнала из файла

Сегментирует аудиосигнал, находящийся в виде wav-файла, на участки с метками, отмечающими речь какого из двух дикторов находится в данном участке.

API функции:

VST_API(vst_res_ptr) processDialog_FromFile(int & resCode, char const* srcFilePath);

resCode адрес для записи завершения работы функции

srcFilePath строка с путем к wav-файлу

Функция возвращает указатель на массив блоков, содержащих структуры VST_results

```
typedef struct vst_res_t
{
int SegmentsAmount;
int SpeakersAmount;
int SignalDuration; // ms
vst_res_segment_ptr Segments; // set of VST-segments
} vst_res_t;
```

Очистка и удаление выходной структуры результатов

API функции:

VST_API(void) deleteData(vst_res_ptr data);

data указатель на массив блоков, содержащих структуры VST_results.

Получение строки с текстом ошибки API функции:

const char* getErrMsg(int code);

code код согласно таблице возвращаемых кодов.

Функция возвращает указатель на строку с текстовой интерпретацией кода ошибки.

5. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

1.2. Модуль 3i Speech Diarization SDK

Требования к входным аудио данным

- возможные источники: WAV-файлы, буфер отсчётов;
- частота дискретизации сигнала: 8 кГц;
- разрядность квантования: 8 бит, 16-бит;
- тип кодирования, если источником является WAV-файл: A-закон, Mu-закон или PCM;
- тип кодирования, если источником является буфер памяти: PCM.

Выходные данные:

указатель на массив блоков, содержащих структуры VST_results

```
typedef struct vst_res_t
{
int SegmentsAmount;
int SpeakersAmount;
int SignalDuration; // ms
vst_res_segment_ptr Segments; // set of VST-segments
} vst_res_t;
```

где SegmentsAmount – количество сегментов, SpeakersAmount – количество дикторов в сегментах, SignalDuration – общая длительность речевого сигнала в мс, SignalDuration – указатель на структуру-описание сегмента:

```
typedef struct vst_res_segment_t
{
int SpeakerID;
float Confidence;
int StartTime; // ms
int EndTime; // ms
} vst_res_segment_t;
```

где SpeakerID – ИД диктора (из базы дикторов), Confidence – достоверность распознавания диктора в данном речевом сегменте, StartTime – начало сегмента в мс от начала файла(буфера), EndTime – конец сегмента от начала файла(буфера)

6. СООБЩЕНИЯ ОПЕРАТОРУ

Для модуля 3i Speech Diarization SDK - не имеется.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

В настоящем документе приняты следующие условные обозначения:

СПО	Специальное программное обеспечение
СУБД	Система управления базами данных
3i Speech Diarization SDK	СПО «3i Speech Diarization SDK»
API	сокр. англ. Application Programming Interface, интерфейс программирования приложений – набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах
DLL	сокр. англ. Dynamic Link Library — динамическая библиотека, позволяющая многократное использование различными программными приложениями.
CPU	сокр. англ. Central Processing Unit электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программ).
DNN	сокр. англ. Deep Neural Network, искусственная нейронная сеть с несколькими скрытыми слоями.
WFST	сокр. англ. Weighted Finite State Transducer, взвешенный конечно-автоматный преобразователь, автомат Мили со взвешенными дугами.
SDK	сокр. англ. Source Development Kit - комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения.
GMM	сокр. англ. Gaussian Mixture Model – статистическая модель плотности вероятности, выраженная суммой нормальных многомерных распределений.

MFCC	сокр. англ. Mel-Frequency Cepstrum Coefficients – представление кратковременного спектра, основанное на линейном косинусном преобразовании логарифмированного амплитудного спектра по мел-частотной шкале.
DTMF	сокр. англ. Dual-Tone Multi-Frequency, двухтональный многочастотный аналоговый сигнал, используемый для набора телефонного номера.
UBM	сокр. англ. Universal Background Model – статистическая модель пространства признаков голоса.
PCA	сокр. англ. Principal Component Analysis – метод сокращения размерности статистических данных.
PCM	сокр. англ. Pulse Code Modulation, импульсно-кодовая модуляция, термин применяется в смысле типа кодирования аудио-сигнала.

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	№ документа	Входящий № сопроводительного документа и дата	Подпись	Дата
	Измененных	замененных	новых	аннулированных					