Text-to-Speech System Documentation

Text-to-Speech System Documentation

Code: *https://github.com/drkhanusa/text2speech*

Model*: khanusa/text2speech · Hugging Face*

System recommendation: GPU RTX3090+

Table of Contents

1. System Overview

The Text-to-Speech system consists of two main components: Triton Inference Server to Handles the core TTS model inference, and FastAPI Service to Provides REST API interface for client applications

Key Features

- Vietnamese text-to-speech synthesis
- Multiple voice styles (male/female, north/south accents)
- Adjustable speech speed
- Support for both streaming and offline modes
- High-performance inference using TensorRT-LLM

2. Component Architecture

2.1. Triton Inference Server

2.1.1. Components

- TensorRT-LLM Backend
- Audio Tokenizer
- Vocoder
- Model Repository

2.1.2. Directory Structure

```
runtime/triton_trtllm/
├── docker-compose.yml
├── run.sh
```

```
├── model_repo/
│   ├── spark_tts/
│   ├── audio_tokenizer/
│   ├── tensorrt_llm/
│   └── vocoder/
└── scripts/
```

2.1.3. Model Pipeline

1. Text input → Audio Tokenizer
2. Audio Tokenizer → TensorRT-LLM Model
3. TensorRT-LLM Model → Vocoder
4. Vocoder → Audio Output

2.2. FastAPI Service

2.2.1. Components

- REST API Interface
- Request Processing
- Audio Processing
- Response Handling

2.2.2. Directory Structure

```
text2speech/
├── text2speech_api.py
├── requirements.txt
└── audio_reference/
    ├── male_north.wav
    ├── female_north.wav
    ├── male_south.wav
    └── female_south.wav
```

3. Installation Guide

3.1 Prerequisites

- NVIDIA GPU with CUDA support
- Docker and Docker Compose
- Python 3.8+
- NVIDIA Container Toolkit

3.2 Step-by-Step Installation

1. Clone the repository:

```
git clone https://github.com/drkhanusa/text2speech.git
cd text2speech
```

2. Install Python dependencies:

```
pip install -r requirements.txt
```

3. Start the Triton server:

```
cd runtime/triton_trtllm
docker compose up
```

Instead of use command “git clone https://github.com/drkhanusa/text2speech.git && cd text2speech/runtime/triton_trtllm”, you can move the text2speech folder into the docker and build NVIDIA Triton server with the same way.

4. Start the FastAPI service:

```
python text2speech_api.py
```

4. Configuration Guide

4.1. Triton Server Configuration

4.1.1. Docker Compose Settings

```
services:
  tts:
    image: soar97/triton-spark-tts:25.02
    shm_size: '1gb'
    ports:
      - "8000:8000"  # HTTP
      - "8001:8001"  # gRPC
      - "8002:8002"  # Metrics
    environment:
      - PYTHONIOENCODING=utf-8
      - MODEL_ID=SparkAudio/Spark-TTS-0.5B
```

4.1.2. Model Parameters

```
# Model Configuration
trt_dtype=bfloat16
MAX_BATCH_SIZE=8
MAX_NUM_TOKENS=32768

# Triton Server Configuration
BLS_INSTANCE_NUM=4
TRITON_MAX_BATCH_SIZE=8
MAX_QUEUE_DELAY_MICROSECONDS=0

# Streaming Parameters
AUDIO_CHUNK_DURATION=1.0
MAX_AUDIO_CHUNK_DURATION=30.0
AUDIO_CHUNK_SIZE_SCALE_FACTOR=8.0
AUDIO_CHUNK_OVERLAP_DURATION=0.1
```

4.2. FastAPI Service Configuration

4.2.1. Server Settings

```
# text2speech_api.py
app = FastAPI()
uvicorn.run(
    app,
    host="0.0.0.0",
    port=13601,
    proxy_headers=True,
    forwarded_allow_ips="*"
)
```

4.2.2. Audio and Text Parameters

Reference Audio Configuration

```
reference_audio_dict = {
    "Nam-Bắc": "audio_reference/male_north.wav",
    "Nữ-Bắc": "audio_reference/female_north.wav",
    "Nam-Nam": "audio_reference/male_south.wav",
    "Nữ-Nam": "audio_reference/female_south.wav",
}
```

Reference Text

- Default reference text: "chào chị, em là linh, chuyên viên tư vấn từ thẩm mỹ viện ngọc dung, em xin phép được hỏi chị một chút về tình trạng da của mình nhé, chị có thể cho em biết là chị có đang gặp phải vấn đề gì về da không ạ"
- This text is used as a style reference for the TTS model
- Should be a natural, conversational Vietnamese text
- Used to maintain consistent voice characteristics

Target Text

- The text you want to convert to speech
- Maximum length: 2048 tokens (approximately 40 seconds of audio)
- Should be in Vietnamese
- Can contain punctuation marks
- Will be automatically split into sentences if longer than the limit

Audio Parameters

- Sample rate: 16000 Hz
- Format: WAV
- Maximum duration: 40 seconds
- Channels: Mono
- Bit depth: 16-bit

Text Processing

```
# Example of text processing in the API
text_lower = request.content_text.lower()
cleaned_text = re.sub(r'\s+', ' ', text_lower).strip()
list_target_text = cleaned_text.split('.')
```

Voice Styles

- Four available voice styles:
- Male North (Nam-Bắc)
- Female North (Nữ-Bắc)
- Male South (Nam-Nam)
- Female South (Nữ-Nam)

Speed Control

- Range: 0.5x to 2.0x
- Default: 1.0x
- Applied after audio generation

5. API Documentation

5.1. Text-to-Speech Endpoint

Endpoint: `POST /t2s`

Request Body:

```
{
    "content_text": "Text to convert to speech",
    "sex": "Nam",  // or "Nữ"
    "region": "Bắc",  // or "Nam"
    "speed": 1.0  // Optional, default is 1.0
}
```

Response:

- Returns a WAV audio file

5.2. Example Usage

```
import requests
url = "http://localhost:13601/t2s"
data = {
    "content_text": "Xin chào, đây là ví dụ về chuyển văn bản thành giọng nói.",
    "sex": "Nam",
    "region": "Bắc",
    "speed": 1.0
}
response = requests.post(url, json=data)
with open("output.wav", "wb") as f:
    f.write(response.content)
```