Text-to-speech in Vietnamese
Fine-tuning Text-to-speech model with VITS

Nguyen Thanh Phat - Intern AI/Data Science at TMA Solutions

Abstract

We use Piper library to finetuning VITS for Text-to-speech (TTS) tasks with different voice in Vietnamese. We also built a Tornado server to deploy TTS model on microservice with Docker. The server uses ONNX model type to infer lightweight and excellent performance.

1. Data preprocessing

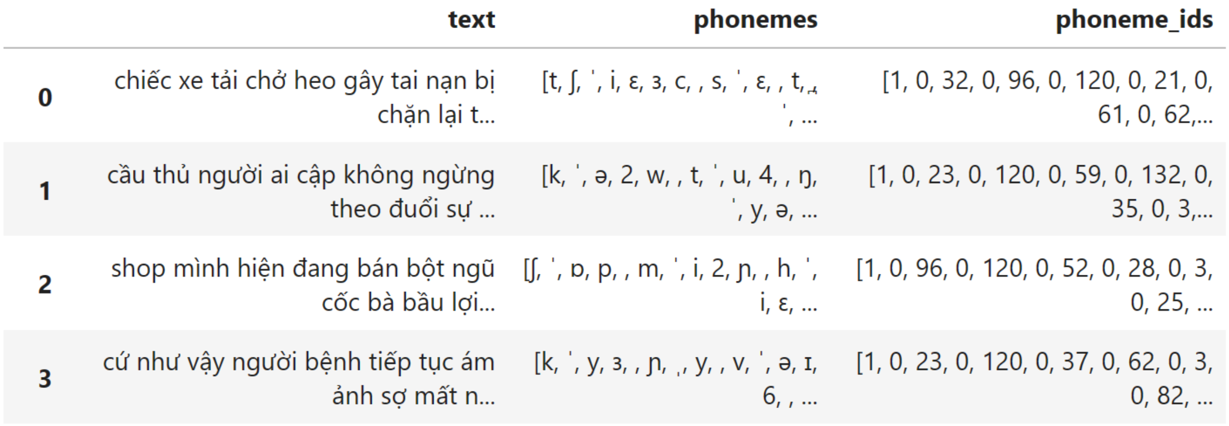
In the finetuning task we use VITS model we use **InfoRe** datasets (2,483 samples) but we only use 800 samples for training and the next 200 samples for evaluation to test how good the model is.

```
        id                                              text
0    13657  sổ hồng riêng mua bán công chứng ngay chủ nhà ...
1    03744  chuối và bơ hạnh nhân cũng được coi là thực ph...
2    13139  anh ấy nghĩ sai làm sai không gãy gánh sự nghi...
3    01294  hy vọng những dự định của thúy sẽ thành công đ...
4    08105  đây là phần nổi còn phần chìm dưới thì khó ai ...
..     ...                                               ...
995  08666  dù việc đi lại khó khăn viết và tiếp thu bài c...
996  13256  và khi đã chọn mua được tượng phật ưng ý chúng...
997  10059  nếu không điều tiết được tốc độ tăng đàn sẽ dẫ...
998  10914  trò chơi sẽ là một kho các câu hỏi và trả lời ...
999  05702  vốn là doanh nhân chuyên nghiệp các nhà đầu tư...

[1000 rows x 2 columns]
```

#Figure 2.1. Structure of InfoRe datasets

For one item of dataset has an **<id>.wav** audio file present for its text. Before training model, we need to convert text to phonemes, the phonemes seem like a multiples symbol present how to pronounce a words or phrase. And we also return the ids of its as a vector for training. We can do this by using `piper_phonemize` library.



	text	phonemes	phoneme_ids
0	chiếc xe tải chở heo gây tai nạn bị chặn lại t...	[t, ʃ, ˈ, i, ɛ, ɜ, c, , s, ˈ, ɛ, , t, ˧ ˈ, ...	[1, 0, 32, 0, 96, 0, 120, 0, 21, 0, 61, 0, 62,...
1	cầu thủ người ai cập không ngừng theo đuổi sự ...	[k, ˈ, ə, 2, w, , t, ˈ, u, 4, , ŋ, ˈ, y, ə, ...	[1, 0, 23, 0, 120, 0, 59, 0, 132, 0, 35, 0, 3,...
2	shop mình hiện đang bán bột ngũ cốc bà bầu lợi...	[ʃ, ˈ, ɒ, p, , m, ˈ, i, 2, ɲ, , h, ˈ, i, ɛ, ...	[1, 0, 96, 0, 120, 0, 52, 0, 28, 0, 3, 0, 25, ...
3	cứ như vậy người bệnh tiếp tục ám ảnh sợ mất n...	[k, ˈ, y, ɜ, , ɲ, ˌ, y, , v, ˈ, ə, ɪ, 6, , ...	[1, 0, 23, 0, 120, 0, 37, 0, 62, 0, 3, 0, 82, ...

#Figure 2.2. Convert text to phonemes.

Before training we need to make sure all the **.wav** audio files has sampling_rate=**22050** because this is the default input audio format of VITS model we use, we can convert it to the right sampling rate by `resample.py` in Piper library. So now the data is ready for training.

2. Training

Training take ~1.5 hour on Google Colab Tesla T4 (16GB) with following hyper-parameters:

Hyper-parameters	**Value**
batch_size	34
epochs	100
learning_rate	auto

#Table 2.1. Hyper-parameters for training

After training we export the model to ONNX format which optimized for inference task. The exported ONNX model weighted **~60.2 MB**, it reduced **~92.8%** of the original model which weighted **~846 MB.** It so good for deploy in low computing devices, such as CPU.

	Original model	Optimized model
Model size	**846 MB**	**60.2 MB**

3. Evaluation

In evaluation task we use MSE and RMSE to compare the model before and after finetuning on the test dataset (200 sample):

	BEFORE finetuning	**AFTER finetuning**
Mean Square Error (lower is better)	0.044785	0.043612
Root Mean Square Error (lower is better)	2.011006	1.977739

#Table 3.1. Metrics in Test datasets

In TTS tasks, the output spectrogram for a given text can be represented in many ways. So, loss functions like MSE and MAE are just used to encourage the model to minimize the difference between the predicted and target spectrograms. The right way to Evaluate TTS model is use with MOS(mean opinion scores) BUT it is a subjective scoring system and we need human resources to do it. So, we use MSE and RMSE for replacement.

4. Conclusion

In summary, our project showcases the potential of fine-tuning the VITS model for Vietnamese TTS tasks and demonstrates the benefits of model optimization for efficient deployment. This work contributes to the development of better text-to-speech systems in the Vietnamese language, with implications for various applications, from accessibility to entertainment and beyond.

5. Code & demo

Video demo: https://www.youtube.com/watch?v=1mAhaP26aQE
Source code: https://github.com/phatjkk/vits-tts-vietnamese