

VITS -Dataset Creation

Table of Contents

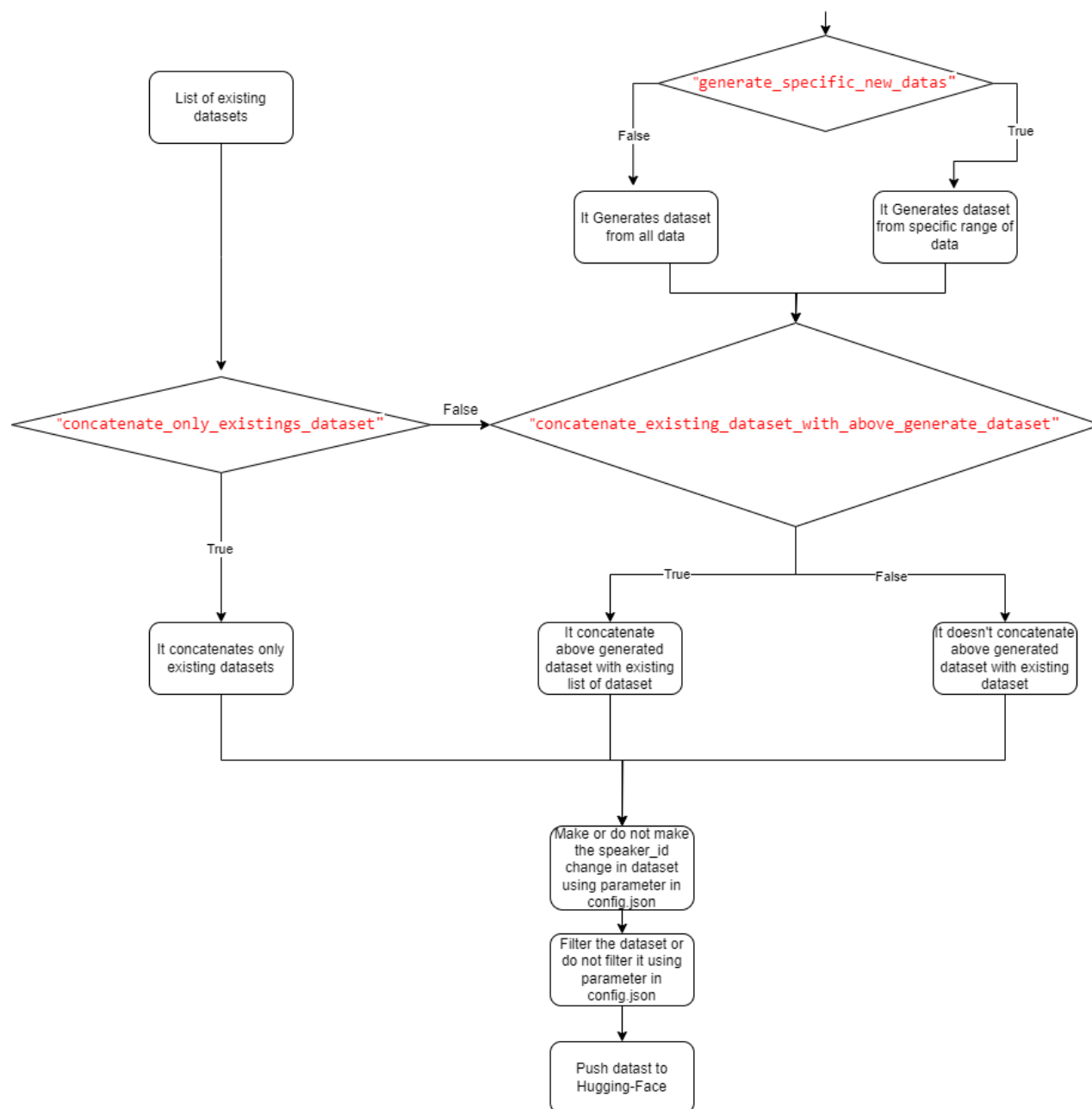
Introduction	2
Flow Chart	3
Draw.IO source :- link	3
Note :- “red” marked text in flowchart are the parameters in config.json file	3
Data Creation Tool	4
Data Format for model	6
Data Pipelining	7
How to run code?	13

Introduction

VITS (Variational Inference with adversarial learning for end-to-end Text-to-Speech) is an end-to-end speech synthesis model that predicts a speech waveform conditional on an input text sequence.

Flow Chart

Draw.IO source :- [link](#)



Note :- “red” marked text in flowchart are the parameters in config.json file

Data Creation Tool

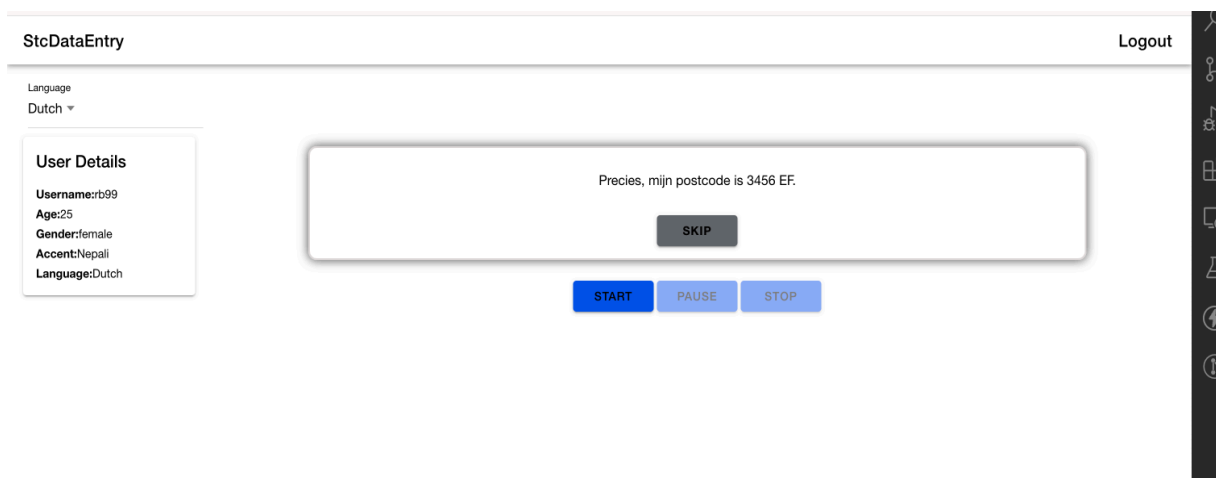
The creation of our data for training the model is being done in different stages:

1. Initially, we used common_voice dataset from the mozilla foundation, easily available in Huggingface which can be accessed through <https://huggingface.co/mozilla-foundation> . We have specifically used **mozilla-foundation/common_voice_16_0** and **mozilla-foundation/common_voice_17_0**.

First of all to get a raw model we used the **mozilla-foundation/common_voice_16_0** and in the later stages to train the raw model further we used **mozilla-foundation/common_voice_17_0**.

2. In the later stages, we have been using a custom dataset. The custom dataset has been created using our data creation tool which can be accessed through link

<https://datacreationtool.procit.com/>



Once we login using our credentials we come across the interface as shown above, where we can select language either Dutch or English. Then we can start to record our voice and read the text given. Once recorded, we can pause or stop the recording, listen to our voice and either redo (in case we do not like the recording) or save it. We also have the option to skip the text and go to another text to record our voice. All we have to do is read the text and save our recording.

All the voice data generated using this tool can be obtained from api [GetDataAPI](https://datacreationtool.procit.com/nodeapi/getData) (<https://datacreationtool.procit.com/nodeapi/getData>) .

3. As the VITS model is unable to speak in character level i.e has difficulty saying A B etc., we have created a dataset from speechgen.io only with data that includes such characters, to train our model. This is an ongoing process as well.

The voice can be manually downloaded from the same site once it is generated. This dataset is then saved locally.

Note: We are using Dutch NL as the language and speaker persona of Hendrika.





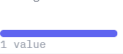








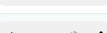
Data Format for model

Normally we need “speaker_id” , “normalized_text” and “audio” columns for the VITS model. The dataset will be generated in the format shown below using our code.

To access the entire dataset go to hugging face and find **procit001/voicesample_from_stc_july_15**

Split (1)
train · 2.37k rows

Search this dataset

speaker_id	speaker_name	accent	text	gender	audio	normalized_text
int64	string · classes	string · classes	string · lengths	string · classes	audio · duration (s)	string · lengths
						
60	sushmita	Nepali	People call me	female		People call me.
60	rikesh	Nepali	Just to confirm, your street name is Main Street, right?	male		Just to confirm your street name is main street right.
60	rikesh	Nepali	That is completely accurate.	male		That is completely accurate.
60	rikesh	Nepali	two	male		Two.
60	rikesh	Nepali	Exactly as I thought.	male		Exactly as i thought.

Data Pipelining

1)Path for Code

svn path :- http://172.20.70.59/svn/svn_procit_devprojects/STC_AI/AI-Module/Datacreation/VITS/trunk

google drive path :- <https://drive.google.com/drive/folders/1vxSvh5YNdacD7FAuvdHyjcolJHMGL6fA>

path in server :- /home/ai001/INT4TB/dataset_creation/dataset_creation_vits/

2)Explanation of Code flow(Algorithm)

- Load config file
- Logging to Hugging-face
- If **'concatenate_only_existings_dataset':true**, concatenate list of existing datasets. Configure speaker_id and apply filters according to different parameters configured in config.json . Else,
- Fetch data created using Data-Creation tool using [api url](https://datacreationtool.procit.com/nodeapi/getData)(<https://datacreationtool.procit.com/nodeapi/getData>)
- Convert API data to dictionary format
- Convert dictionary to dataset
- Rename the columns in Dataset as per our requirements.
- Filter rows in Dataset.
 - We can filter the rows in Dataset by setting **'should_filter'** to **'true'** and setting **'should_filter_speaker_name'** to **'true'** and by giving param value in list like **"should_filter_speaker_name_param": ["saskia001","conny001"]** in config.json file to filter dataset according to "saskia001" and "conny001" user.
 - Similarly we can filter the dataset on the basis of 'gender', 'accent', 'language', 'speaker_name', and specific word in the 'text' column as per our requirements.
- Remove unnecessary column from dataset
- Now, push the dataset to Hugging-Face. The name for dataset will be according to the parameter **"dataset_name"** in config.json
 - The generated dataset has column name 'audiopath', which contains the path of the audio file. But we need an actual audio array in the dataset for Model trains. For this we download the audio from the provided url and then add these downloaded audio to the dataset.
 - Due to limitation of resources, we can't convert all the sound files from urls and add these to the dataset at once. To resolve this problem, we do this process by making a batch of data.
 - Normally, we set **"batch_size":1000** in the config.json file. We can use different batch sizes which can be handled by our system. If there are 6500 numbers of data and we set

- “batch_size”:1500 then it will make datasets which will have 1500 data in each dataset and another dataset for remaining data. In total 5 dataset will be generated where the first four dataset will have 1500 data in each and remaining 500 data will be in the fifth dataset.
 - The names for generated datasets will be according to parameter “dataset_name” in config.json with integer number appended at last. for example “dataset_name_0”, “dataset_name_1” and so on.
- Make a Batch of dataset and push to Hugging-Face. Batch size is set in config.json as param “batch_size”.
- Again load the batched datasets and add a new column ‘audio’ and add sound array to that column.
- Normalize the ‘text’ and make a new column name ‘normalized_text’. We use ‘normalized_text’ instead of ‘text’ for model trains. Normalization is done to remove unnecessary symbols in text.
 - It converts number to dutch text
 - It capitalizes the character following a comma (“ , ”) or period (“ . ”)
 - It removes special characters except for “ @ ”
 - It capitalizes the first word.
 - It adds a full stop at the end of the sentence.
- Again Push all the dataset having ‘audio’ and ‘normalized_text’
- Now, Concatenate all the dataset. All the concatenated dataset now push in the name “dataset_name” in config.json . This is the generated dataset using datas from a data-creation tool.
- The dataset generated above can be either a dataset for the entire data or a dataset for the newly received data on the basis of “generate_specific_new_datas” parameter in config.json .If “generate_specific_new_datas”:false, dataset will be created from all the data, and if “generate_specific_new_datas”:true, then specific range of dataset will be created.
 - If we set “generate_specific_new_datas”: false , the dataset will be created from the entire data.
 - If we set “generate_specific_new_datas”:true, then it is necessary to provide parameter values for “start_end_value” in config.json
 - “start_end_value” should be an array of integers. e.g. “start_end_value”:[4500,5000] . It is used to get only a specific range of data.
 - Normally we use [api_url](https://datacreationtool.procit.com/nodeapi/getData) (https://datacreationtool.procit.com/nodeapi/getData)
 - To get specific range of data we use [api_url](https://datacreationtool.procit.com/nodeapi/getData?starting=4500&end=5000) (https://datacreationtool.procit.com/nodeapi/getData?starting=4500&end=5000)
 - **NOTE** :- api_url has been changed. previously api was [api_url](https://psa-dev.kcmsecurity.eu/stcai-datasetcreationapi/getData) (https://psa-dev.kcmsecurity.eu/stcai-datasetcreationapi/getData) to get the data.
- If ‘concatenate_existing_dataset_with_above_generate_dataset’:true, then the list of existing dataset[‘train’] will be concatenated with generated dataset[‘train’]. Else, if set to false then only the generated dataset will be pushed to the Hugging-Face .
- Change on column of dataset

- There are different columns in the pushed dataset like “speaker_id”, “speaker_name”, “accent”, “text”, “audio”, “gender”, and “normalized_text”. For now we can just change the column value for the “speaker_id” column.
- If we set “make_change_on_columns”:true , set “make_speaker_id_same”:true we set “make_speaker_id_same” to some integer value then for the whole dataset there will be the same “speaker_id”.
- If we set “make_change_on_columns”:true , set “make_speaker_id_same”:false then for the whole dataset there will be a unique “speaker_id” for different “speaker_name”. The unique “speaker_id” will be the integer number starting from 1.
- If we set “make_change_on_columns”:false then “speaker_id” value will be same as the original
- Apply filter according to parameter set in config.json
- Push to Hugging-face using name “dataset_name” defined in config.json

3)Explanation of config.json file parameters

Note:- It is necessary to set proper values for the parameters in the config. json file to run the program properly.

Parameter	Type	Example	Description	Remarks
“huggingface_hub_token”	String	"huggingface_hub_token":"hf_jYJGLnrVeNnJbqHgvdvUGHYnsBmctrW"	“token” of the Hugging-face where you want to push the dataset	
"dataset_name"	String	"dataset_name":"procit002/test"	This name will be set to the dataset created.	dataset name must be a full name like “"procit002/test"
"batch_size"	Integer	"batch_size":1000	Convert entire dataset into number of dataset having row equals to 1000 in each	Normally we use “batch_size”:1000 , we can increase the value of it until our server can handle it.
"concatenate_only_existings_dataset"	Boolean	"concatenate_only_existings_dataset":false	If it is set to true, then only list of existing dataset will concatenated	We must set it “false” if we want to generate a new dataset for data

				from a data-creation tool.
"make_change_on_columns"	Boolean	"make_change_on_columns":true	If we want to make change on column of dataset , we set it to true	
"make_speaker_id_same"	Boolean	"make_speaker_id_same": true	If we want to make speaker_id the same for all , we set it to true.	To implement this "make_change_on_columns" must be true.
"make_speaker_id_same_value"	String	"make_speaker_id_same_value":"5"	It will be set on "speaker_id" column	set it's value in the form of string
"generate_specific_new_datas"	Boolean	"generate_specific_new_datas":true,	If we want to make a dataset for the entire data set it to false. If we want to make dataset for specific range of data then set it to true	When we set "generate_specific_new_datas":true, we also must set the value for "start_end_value" parameter
start_end_value	Array of integer	"start_end_value":[1000, 1500],	It is used to get a specific range of data from api. The first value in the array is the starting value whereas the second value is the end value for api url.	It should have two integer value in array
"concatenate_existing_dataset_with_above_generate_dataset"	Boolean	"concatenate_existing_dataset_with_above_generate_dataset":false,	If we want to concatenate existing datasets with above generated dataset then, we must set it "true"	

"existing_dataset_name"	Array of String	"existing_dataset_names":["procit002/test_1","procit002/test_2"],	It is the list of dataset that already exists in hugging-face.	Note:- The columns and features of existing and newly generated dataset should be the same to concatenate them.
"should_filter"	Boolean	"should_filter":true	This is set to true if we want to filter the dataset according to the different parameter like 'gender', 'accent', 'language', 'speaker_name', and 'word' in 'text' column	
"should_filter_gender"	Boolean	"should_filter_gender":true	set it true if we want to filter gender according to provided array of string in "should_filter_gender_param"	
"should_filter_gender_param"	Array of string	"should_filter_gender_param":["female"]	It is used to filter the 'gender' in the dataset. "should_filter_gender_param":["female"], it will generate the dataset only containing 'gender' = 'female'	we can know the different gender types from get api or existing dataset
"should_filter_accent"	Boolean	"should_filter_accent":false,	set it true if we want to filter accent	we can know the different accent types from get api

				or existing dataset
"should_filter_acc ent_param"	Array of string	"should_filter_accen t_param":["Nepali","Dutch", ["English"],	put the list of accent types to filter dataset	
"should_filter_lan guage"	Boolean	"should_filter_language" :false	If we want to create a dataset according to the language then set it true.	What are different languages available in data can be found from get_api_url or existing dataset.
"should_filter_lan guage_param":["nl "]	Array of string	"should_filter_language _param":["nl"]	put the list of languages to filter dataset accordingly	
"should_filter_spe aker_name"	Boolean	"should_filter_speaker_ name":true,	If we want to filter the dataset on the basis of 'speaker_name' then we set it to true.	What are different speaker_name available in data can be found from get_api_url or existing dataset.
"should_filter_spe aker_name_param "	Array of string	"should_filter_speaker_ name_param":["saskia00 1","conny001"],	put the list of speaker_name to filter dataset accordingly	
"should_filter_sen tence_containing_ word"	Boolean	"should_filter_sentence_ containing_word":true,		
"should_filter_sen tence_containing_ word_param"	Array of string	"should_filter_sentence_ containing_word_param ":["huisnummer"]	It helps to create a dataset where the "text" column contains specific words provided in Array.	

How to run code?

- Make environment
 - `python3 -m venv env`
- Install all the requirements . There is a 'requirements.txt' file which contains all the required libraries and packages.
 - `pip install -r requirements.txt`
- Run the code
 - `python3 main.py -- config config.json`