Multilingual Aspects of the Swarajability Portal

A Project Report Submitted
in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Vibhanshu Jain
(Roll No. CS19B1027)

ಭಾರತೀಯ ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಸಂಸ್ಥೆ ರಾಯಚೂರು
भारतीय सूचना प्रौद्योगिकी संस्थान रायचूर
Indian Institute of Information Technology Raichur

to

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY RAICHUR
RAICHUR - 784135, INDIA

April 2023

DECLARATION

I, **Vibhanshu Jain (Roll No: CS19B1027**, hereby declare that, this report entitled **"Multilingual Aspects of the Swarajability Portal"** submitted to Indian Institute of Information Technology Raichur towards the partial requirement of **Bachelor of Technology** in **Computer Science and Engineering Department**, is an original work carried out by me under the supervision of **Dr. Maunendra Sankar Desarkar** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. I have sincerely tried to uphold academic ethics and honesty. Whenever a piece of external information or statement or result is used then, that has been duly acknowledged and cited.

Raichur - 784 135 **Vibhanshu Jain**
April 2023

CERTIFICATE

This is to certify that the work contained in this project report entitled **"Multilingual Aspects of the Swarajability Porta;"** submitted by **Vibhanshu Jain** (**Roll No: CS19B1027**) to Indian Institute of Information Technology Raichur towards the partial requirement of **Bachelor of Technology** in **Computer Science and Engineering Department** has been carried out by him under my supervision and it has not been submitted elsewhere for the award of any degree.

Hyderabad - 584 134
May 2023

Dr. Maunendra Sankar Desarkar
Project Supervisor

ABSTRACT

Swarajability Portal is a job portal for persons with disabilities. The project objective is to support this portal with AI, in terms of job recommendation on the basis of disability, job requirement, location, and other details about the candidates and the job description and a very smooth and accurate translation of the job description into multiple Indian languages. The project also extends towards the transliteration of the converted job description into the male and female voice in the local languages. For the translation and transliteration, we used the data set and the model provided by Ai4Bharat, which is an organization working on the NLP for Indian Languages.

The final result of the project is providing the APIs for the translation, and transliteration of the input text along with a working prototype of the job portal in multiple Indian languages. The whole system is lightweight and very easy to set up in any new environment.

Contents

Contents v

List of Figures vii

List of Tables viii

1 Introduction 1
- 1.1 Motivation behind this project . . . 1
- 1.2 Swarajability Portal . . . 1
- 1.3 Enhancing Accessibility with the Help of AI . . . 2

2 Introduction to text translation 3
- 2.1 Introduction . . . 3
- 2.2 Basic Understanding of text-translation with AI . . . 3
- 2.3 Neural Machine Translation (NMT) . . . 4
- 2.4 Some available models / services . . . 5
  - 2.4.1 Google Translate . . . 5
  - 2.4.2 Deepl . . . 5
  - 2.4.3 Microsoft Translator . . . 5
  - 2.4.4 Taia . . . 5
  - 2.4.5 Reverso . . . 5
  - 2.4.6 Memoq . . . 5
  - 2.4.7 Systran . . . 5
  - 2.4.8 Amazon Translate . . . 5
  - 2.4.9 ChatGPT . . . 5
- 2.5 Performance on Indian Languages . . . 5

3 AI4Bharat - IndicTrans 6
- 3.1 Artificial-Intelligence For Bhārat . . . 6
- 3.2 IndicTrans . . . 6

3.2.1 Samanantar Dataset . . . 7
3.2.2 Currently released models . . . 7
3.2.3 BenchMark . . . 7
3.3 API Environment . . . 8
3.3.1 API Information . . . 8
3.3.2 Enviroment Requirements . . . 8

4 Text to Speech (TTS) 9
4.1 Revolutionizing text-to-speech synthesis with AI and deep learning 9
4.2 Some recent TTS Models . . . 10
4.3 Indic TTS . . . 10
4.4 Model Details provided by AI4Bharat . . . 10

5 Before you copy and paste 12
5.1 Why this template? . . . 12
5.2 How much *LaTeX* you need? . . . 12
5.3 Where to start in this template? . . . 12
5.3.1 `config/packages.tex` . . . 12
5.3.2 `config/options.tex` . . . 13
5.3.3 `config/colors.tex` . . . 13
5.3.4 `config/theoremstyle.tex` . . . 13
5.3.5 `config/mathletters.tex` . . . 13
5.4 Adding content . . . 13
5.5 Front Matter . . . 13
5.6 Citations and References . . . 13
5.7 Info . . . 14

Appendices 17

A Results from Measure Theory 17

Bibliography 19

List of Figures

4.1 Overall Design 9
4.2 Overall Design 11

5.1 Poor boy needs money. Help him by sending your charitable donations 15

List of Tables

3.1 List of Languages 7

Chapter 1

Introduction

1.1 Motivation behind this project

This project endeavors to provide an exceptional recommendation and text translation system to the esteemed Youth4Job Organization. This organization has been at the forefront of helping physically challenged individuals in securing employment opportunities. The Swarajability Portal is a remarkable platform designed to cater to the unique needs of these individuals. However, the platform's English language-centric design may pose a challenge for people from diverse linguistic backgrounds.

To address this challenge, our project seeks to integrate a cutting-edge system that will enable users to seamlessly access the platform in their preferred language. The system's innovative features will also facilitate audio descriptions of job details and requirements, thereby providing equitable access to the platform for visually impaired individuals.
In essence, our system's implementation will undoubtedly revolutionize the Swarajability Portal, allowing the organization to reach a more diverse audience and enhance its impact on society.

1.2 Swarajability Portal

The Swarajability Portal currently offers users and employers a login feature, which allows users to create their profiles while specifying their physical disabilities, location, and other relevant geographical information. Additionally, users are required

to provide comprehensive educational and skill-related details, including any training they have received in the past.

Employers can easily add job listings to the portal, which users can then apply for. However, the platform is currently only available in English, which may hinder access for individuals with diverse linguistic backgrounds.

Our project aims to address this limitation by implementing a state-of-the-art multi-language feature that will enable users to access the platform in their preferred language. This innovative feature will enhance the user experience and make the platform more inclusive for people from different linguistic backgrounds.

Moreover, our system will continue to offer the login feature, enabling users to create detailed profiles that capture their unique attributes and skills, making them more marketable to potential employers. Employers, on the other hand, can easily add job listings, creating a seamless process for users to search and apply for available positions.

In summary, our project's implementation will not only enhance the user experience but also improve the platform's overall efficiency and effectiveness in connecting physically challenged individuals with employment opportunities.

1.3 Enhancing Accessibility with the Help of AI

Our project focuses on implementing a cutting-edge system that will enable the Swarajability Portal to support multiple languages. We plan to achieve this by providing an API that developers of the portal can utilize to translate text into various languages seamlessly.

Our system will also provide a base 64 encoded format of the translated voice, enhancing accessibility for visually impaired individuals. To achieve these results, we utilized the advanced models and data sets provided by Ai4Bharat, an organization that specializes in Indic language translation and transliteration.
We optimized and exposed the API in a user-friendly manner, making it easily accessible for developers to integrate the system into the Swarajability Portal. Our solution is packaged as a single module, streamlining the integration process and ensuring that the new feature is seamlessly incorporated into the existing platform.

Chapter 2

Introduction to text translation

2.1 Introduction

Text translation is the act of transforming written content into a different language while preserving the original intent, meaning, and style. This process entails ensuring that elements such as humor, anger, slang, scientific terminology, and jokes are accurately conveyed. Translation can be performed either by human translators or by utilizing AI models.

While human translation methods can produce high-quality translations, they require significant manpower and may not be practical for lengthy documents or large volumes of content. As a result, AI models have become increasingly popular as they offer an efficient and cost-effective solution to address translation needs.

2.2 Basic Understanding of text-translation with AI

Languages can be viewed as vast datasets consisting of words that are organized and governed by unique rules and grammar structures. Translating words between languages is relatively straightforward, but simply replacing words won't suffice because languages have distinct sentence structures and convey sentiments and emotions in different ways.

Therefore, a higher level of translation model is required to accurately convey the intended meaning and context of a sentence, taking into account nuances such

as cultural references, idioms, and colloquialisms. This level of sophistication is necessary to ensure that the translation captures not just the words, but the meaning and tone of the original text, as well.

Current AI translation systems employ a technology called Neural Machine Translation (NMT), which was first developed by Google. This approach involves analyzing patterns in large volumes of text in both the source and target languages, which allows the system to identify and learn the relationships between words and phrases.

Using this knowledge, the AI translation model can generate translations that accurately reflect the meaning and tone of the source text, while taking into account the unique grammatical structures and nuances of the target language. This technology has significantly improved the accuracy and efficiency of machine translation, making it a valuable tool for individuals and organizations that require high-quality translations in a timely and cost-effective manner.

2.3 Neural Machine Translation (NMT)

Neural Machine Translation (NMT) is an advanced technique used for translation. This method uses artificial neural networks for improving performance, which involves training a deep learning model on a large dataset containing both the source and target languages.

NMT models are consisting of an encoder and a decoder which encode the input text and generate the translation in the target languages respectively. This kind of architecture helps to handle long and complex sentences.

2.4 Some available models / services

2.4.1 Google Translate

2.4.2 Deepl

2.4.3 Microsoft Translator

2.4.4 Taia

2.4.5 Reverso

2.4.6 Memoq

2.4.7 Systran

2.4.8 Amazon Translate

2.4.9 ChatGPT

2.5 Performance on Indian Languages

While many online translation services and models are effective for a wide range of languages, they may not be as accurate for Indian languages. The limited amount of training data available for Indian languages can make it difficult to develop highly accurate translation models. Therefore, it is important to have a large and diverse dataset specifically designed for Indian language translation in order to train effective translation models for Indic languages.

In upcoming chapters, we will discuss a similar model that is specifically designed for the translation of Indian languages and attempts to address this challenge with a large and carefully curated dataset.

Chapter 3

AI4Bharat - IndicTrans

3.1 Artificial-Intelligence For Bhārat

AI4Bharat is an endeavor led by IIT Madras that is dedicated to creating open-source AI models for various Indian languages. The primary goal of this organization is to enhance and advance AI capabilities specifically for Indian languages by leveraging large datasets.

AI4Bharat concentrates on various areas of AI, such as Machine Translation, Machine Transliteration, Speech Recognition, Language Understanding, Language Generation, and sign language recognition.

For the purpose of text translation, we are using the IndicTrans Model provided by AI4Bharat, which we will be discussing in the next section.

3.2 IndicTrans

IndicTrans is a Transformer-4x (434M) NMT model trained on a very big dataset and opensource. The dataset used in the order to train the dataset is released as the "Samanantar" dataset by the organization.

The model used by AI4Bharat employs a single script approach, which involves converting all Indic data to the Devanagari script. This approach facilitates better lexical sharing among languages, which is beneficial for transfer learning. It also helps to avoid fragmentation of the subword vocabulary among Indic languages and enables the use of a smaller subword vocabulary.

3.2.1 Samanantar Dataset

The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages

The dataset contains a total of 49.7 million sentence pairs between English and 11 Indic languages (from two language families). It contains parallel language sentences ranging from 141K pairs between English-Assamese to 10.1M pairs between English-Hindi. About 37.4M pairs are mined by different mining methods explained in the paper, with only 12.4 M existing resources, giving the new dataset which is about 3 times of the existing one.

https://arxiv.org/abs/2104.05596

Table 3.1: List of Languages

Language
Assamese (as)
Hindi (hi)
Marathi (mr)
Tamil (ta)
Bengali (bn)
Kannada (kn)
Odia (or)
Telugu (te)
Gujarati (gu)
Malayalam (ml)
Punjabi (pa)

3.2.2 Currently released models

They released two models, indic-english and en-indic which support 11 Indian Languages.

3.2.3 BenchMark

The IndicTrans model has been subjected to evaluations on various benchmark datasets, namely WAT2021, WAT2020, WMT (2014, 2019, 2020), UFAL, PMI (a subset of PMIndia dataset curated by the evaluators for Assamese), and FLORES.

The results demonstrate that the model outperforms all publicly available open source models, while also surpassing commercial systems like Google and Bing Translate on most datasets. Furthermore, it delivers competitive performance on the FLORES benchmark.

3.3 API Environment

From the organization Ai4Bharat, we get the trained model which is trained on the large Samantha data set and which can be used for Text translation. The next step is to create a very easily usable API environment that can be used by developers in order to get the text translation.

As the developers are not expected to be very friendly with the Python environment, the next is to pack the complete codebase with all the environment creation details so that the developers can deploy it very easily on their systems. The API structure and example are shared using The Postman collection. Whereas the deployment details are provided in the subsequent chapter.

3.3.1 API Information

Supported Languages

Single Language Translation

Translation to all languages

3.3.2 Enviroment Requirements

Chapter 4

Text to Speech (TTS)

4.1 Revolutionizing text-to-speech synthesis with AI and deep learning

Speech synthesis is a specialized area of artificial intelligence that focuses on generating human-like speech. A typical application of this technology involves utilizing a text-to-speech system, which transforms written text into audible speech. A standard TTS model works in 4 blocks, which are Preprocessor, Encoder, Decoder, and Vocoder. All these blocks have specific tasks allocated to them. The decoder and vocoder are a text analysis module that converts text to linguistic features, and an acoustic model that converts linguistic features to acoustic features. Firstly the preprocessor performs the task of processing the data given as input in which it tokenizes the sentences into words and individual phrases, then segmenting the input into phenomes. Here, phenomes refer to the smallest sound unit which can distinguish words in a given language. The encoder converts the input of Linguistic features (Phonemes) and outputs an n-dimensional embedding which is known as the latent feature. It lies between the Encoder and Decoder. The decoder is used

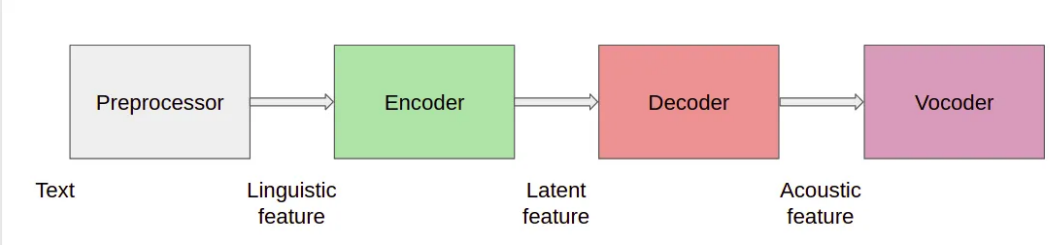

Figure 4.1: Overall Design

to convert information embedded in the Latent processed feature to the Acoustic feature. In the next step, the decoder converts the information embedded in the Latent processed feature to the Acoustic feature.
Vocoders convert the Acoustic feature to the waveform.

4.2 Some recent TTS Models

4.3 Indic TTS

The Indic TTS models are introduced in the paper titles, **TOWARDS BUILDING TEXT-TO-SPEECH SYSTEMS FOR THE NEXT BILLION USERS** by AI4Bharat , Indian Institute of Technology Madras (IITM), India Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), UAE Microsoft Research, India. The paper evaluates different choices of acoustic models, vocoders, supplementary loss functions, training schedules, and speaker and language diversity for Dravidian and Indo-Aryan languages. As the result, the paper shared that monolingual models with FastPitch and HiFi-GAN V1, trained jointly on male and female speakers to perform the best. With the same model, they trained it for 13 languages and the results were better than existing models measured by mean opinion scores. Later they open-sourced all the models.

4.4 Model Details provided by AI4Bharat

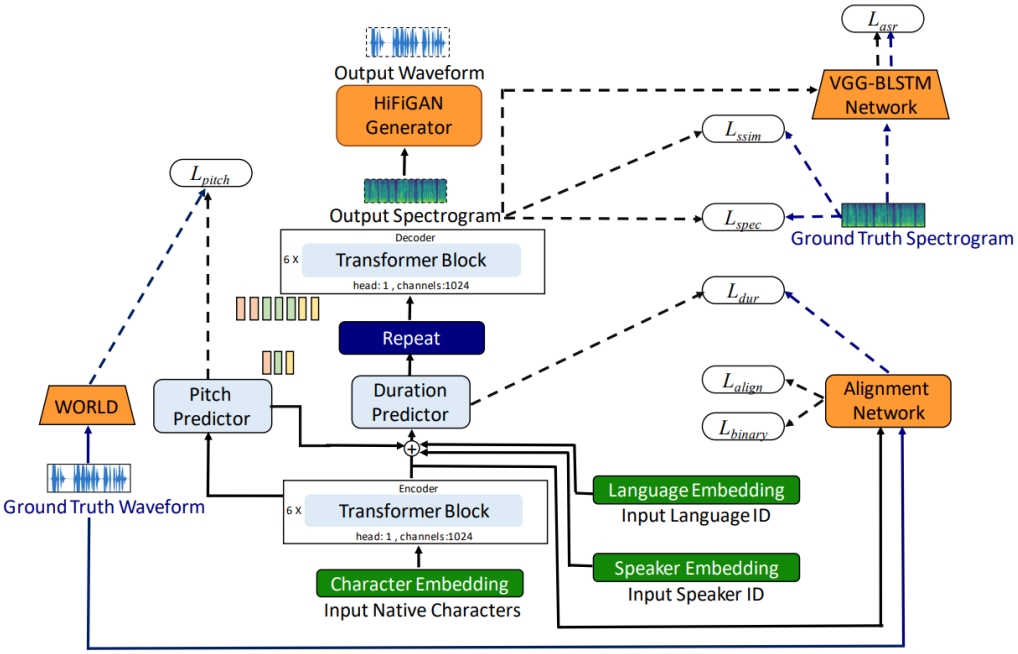

Figure 4.2: Overall Design

Chapter 5

Before you copy and paste

5.1 Why this template?

This template is designed with only one thing in mind. The readability of the script. One of the hardest part in debugging any code, not just *LaTeX* is squashing bugs. With a good structure to your working directory, you'll be able to read and understand what everything means and where to make changes. Without further due let us start

5.2 How much *LaTeX* you need?

I am under the impression that you might know some *LaTeX* scripting. Don't worry if you don't, there are plenty of YouTube videos out there. Get just the basics like how to make a simple text document, some math symbols, inline equations, block equations, aligning equations etc. `https://latex.js.org/playground.html` is an online playground which will give you a quick introduction.

5.3 Where to start in this template?

The directory is setup in a way that almost all the settings are inteh `config` directory. If you open `main.tex`, the first thing included is the `config/packages.tex`.

5.3.1 `config/packages.tex`

This file imports some of the essential packages that are required. I've added what some of these packages do as comments next to them. Uncomment/add

the required packages here. For more details about these packages go to https://www.ctan.org/pkg/*packagename* and read the documentation there.

5.3.2 config/options.tex

The next thing imported by `main.tex` is `config/options.tex`. This file contains all the settings for the packages imported from `config/packages.tex`. If you want to specify further options, put all of them into this file. Refer the documentation at https://www.ctan.org for specific options of the packages.

5.3.3 config/colors.tex

This file contains color settings.

5.3.4 config/theoremstyle.tex

This file contains the settings to stylize definitions and theorems for your document.

5.3.5 config/mathletters.tex

This file contains some shortcuts which helped me typeset math equations.

5.4 Adding content

The chapters of the report is supposed to be in the folder `01Chapters`. Make a tex file for each chapter as it simplifies the content.

5.5 Front Matter

Edit the files in `00Intro` with your details.

5.6 Citations and References

Cite whatever you want with `autocite` like [Rud87, Theorem 3.14 p. 69] and refer things with `autoref`, its way better than `ref`. You can just refer things like section 5.1 and don't have to do section 5.1(Look at the code to see what I mean)

The bibliography file is set to `02End/math.bib`, using `bibsource` in line 8 of `main.tex`. Edit the `.bib` file adding your citations. The citationstyle can also be changed. Refer `biblatex` documentation for this.

Remember to run `biber` if you are working in your local system and not overleaf.

5.7 Info

Fork me on GitHub and contribute to the project.

Figure 5.1: Poor boy needs money. Help him by sending your charitable donations

Appendices

Appendix A

Results from Measure Theory

Here'll we'll discuss some important results from measure theory which are essential for our subject. We already defined what is an L^p function in a given space at **??**.

Proposition A.0.1. *Continuous functions in $\mathbb{T}$, (refer **??**) are dense in $L^p(\mathbb{T})$ for $1 \leq p < \infty$.*

Proof. This is a direct consequence of [Rud87, Theorem 3.14 on p. 69]. Since $\mathbb{T}$ is identified with $[0, 1)$, all continuous functions in $\mathbb{T}$ are compactly supported. □

Proposition A.0.2. *Let $C_c(\mathbb{R})$ be the set of all compactly supported continuous functions in $\mathbb{R}$, then $C_c(\mathbb{R})$ is dense in $L^p(\mathbb{R})$. This is [Rud87, Theorem 3.14 p. 69].*

Proposition A.0.3. *$L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is dense in $L^2(\mathbb{R})$.*

Proof. Let $C_c(\mathbb{R})$ denote the set of compactly supported continuous functions in $\mathbb{R}$. Since every function is continuous and compactly supported, $C_c(\mathbb{R}) \subset L^p(\mathbb{R})$ for all $1 \leq p < \infty$. Therefore $C_c(\mathbb{R}) \subset L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. Then by [Rud87, Theorem 3.14 on p. 69] $C_c(\mathbb{R})$ is dense in $L^2(\mathbb{R})$ and therefore $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ is dense in $L^2(\mathbb{R})$. □

If you follow the proof of the above theorem close enough, you'll see that we can make a stronger claim. Since, $C_c(\mathbb{R}) \subset L^p(\mathbb{R})$ for all $1 \leq p < \infty$,

$$C_c(\mathbb{R}) \subset \bigcap_{1 \leq p < \infty} L^p(\mathbb{R})$$

and therefore again by [Rud87, Theorem 3.14 on p. 69], $\bigcap_{1 \leq p < \infty} L^p(\mathbb{R})$ is dense in $L^q(\mathbb{R})$ for all $1 \leq q < \infty$. We will state the generalization of this as a seperate result.

Proposition A.0.4. *If $f \in L^p(\mathbb{R})$, then*

$$\lim_{\delta \to 0} \int_{\mathbb{R}} |f(x+\delta) - f(x)|^p \, dx = 0$$

Proof. Since $f \in L^p(\mathbb{R})$, for every $\epsilon > 0$ there exist an X such that

$$\left(\int_{|x|>X-1} |f(x)|^p \, dx \right)^{\frac{1}{p}} < \epsilon$$

Therefore by Minkowski's inequality, for $\delta \leq 1$,

$$\left(\int_{\mathbb{R}} |f(x+\delta) - f(x)|^p \, dx \right)^{\frac{1}{p}} \leq \left(\int_{-X}^{X} |f(x+\delta) - f(x)|^p \, dx \right)^{\frac{1}{p}} + 2\epsilon$$

Now since $C_c(\mathbb{R}) \cap L^p(\mathbb{R})$ are dense in $L^p(\mathbb{R})$ by Proposition A.0.2, there exists a $g \in C([-X-1, X+1]) \cap L^p([-X-1, X+1])$ such that

$$\|f - g\|_{L^p([-X-1,X+1])} = \left(\int_{-X-1}^{X+1} |f(x) - g(x)|^p \, dx \right)^{\frac{1}{p}} < \epsilon$$

Then by Minkowski's inequality

$$\left(\int_{-X}^{X} |f(x+\delta) - f(x)|^p \, dx \right)^{\frac{1}{p}} \leq \left(\int_{-X}^{X} |g(x+\delta) - g(x)|^p \, dx \right)^{\frac{1}{p}} + 2\epsilon$$

Now since g is continuous in a compact space $[-X-1, X+1]$, it is uniformly continuous. and since ϵ does not depend on the therefore as $\delta \to 0$ the above integral tends to 0. Hence the proof. □

Bibliography

[Rud87] Walter Rudin. *Real and complex analysis*. McGraw-Hill, 1987, p. 483. ISBN: 978-0-07-054234-1.