

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will ﬁnd the receipt information regarding your submission.

The ﬁrst page of your submissions is displayed below.

Submission author: Assignment title: Submission title: File name:

File size: Page count: Word count:

Character count: Submission date:

Submission ID:

Dikson Rajbanshi Initial Project Plan

Initial Project Plan 77202796 Initial\_Project\_Plan\_77202796.docx 32.11K

6 1,038 5,764

10-May-2021 09:28PM (UTC+0545)

1582789865

|  |  |  |
| --- | --- | --- |
| **BSc (Hons) Computing Course 2020/21**  **Level 6 Production Project** | | |
| **Name:** Dikson Rajbanshi | **Student I.D.:** 77202796 | |
| **Course:** BSc (Hons) Computing | **Supervisor’s Name:** | |
| **Final Project Individual** | | |
| **Title of my Project:** Automatic Speech Recognition of Nepali Language to Devanagari Script | | |
| **Aim of my Project:**  The aim of the project is the development of a Speech-To-Text technology for Nepali language. This will allow for further Natural Language Processing and development of Artificial Intelligence in Nepali language/ Devanagari script. | | |
| **Objectives of my Project:**   * To do research on Automatic Speech Recognition methods and their implementation * To create an ASR engine for Nepali speech into written form in Devanagari Script * To further explore the usage of this technology in Natural Language Processing for Nepali speech | | |
| **Specification of my Product:**   |  |  | | --- | --- | | Functional Requirement | MoSCoW | | Convert Nepali Speech into Written form in Devanagari script | M | | Have a Word Error Rate i.e., error percentage in converting speech to text of less than 10% | M | | Be able to be trained initially with minimum amount of speech data | S | | Be able to be fine-tuned and retrained later using newer data | S | | APIs to access the model for use | S | | Store the audio recorded to be reviewed and reused for training of the model | C | | Convert live stream of audio to text | W | | Generate confidence score for the transcribed text | W |  |  |  | | --- | --- | | Non-Functional Requirement | MoSCoW | | Be able to derive transcription(text) from a recorded file or from the microphone | M | | Developed using open-source software and packages | S | | Machine/ platform-independent | S | | Product must be user-friendly | C | | Properly documented product | W | | | |
| **Research:**  Automatic Speech Recognition (ASR) is a technology where a machine tries to understand patterns and recognize inputs in speech form/ waveform of the speech and convert it to written text (O’Shaughnessy, 2008). Since the development of the concept of ASR, it has always been regarded as an important communication for human-machine interaction. With the development of Internet of things (IoT) in recent years, this has become a primary means of communication for humans to interact with machines (Gong, Li, Deng and Haeb-Umbach, 2016). The limit to a better ASR engine has always been the limited amount of labelled data especially in a language like Nepali. Wav2Vec uses a multi-layer convolutional neural network with unsupervised pre-training on unlabeled data for improving supervised training performance through raw audio representation since unlabeled data are easier to collect (Schneider et al., 2019). Wav2Vec 2.0 uses self-supervision learning with fine-tuning on labelled data which outperforms previously best semi-supervised methods like DeepSpeech 2 (Baevski et al., 2020). | | |
| **Evaluation:**  The evaluation of the final product from this research would be based on the fulfillment of the requirements and specifications as mentioned. The assessment and analysis of the final model would be based on the calculated Word Error Rate (WER) which is the accuracy for ASR engines. The model would be tested for the training accuracy/WER and then testing WER will be checked. This model evaluation will have both the quantitative and qualitative analysis with the WER calculation and model testing on real world application. The model analysis/ assessment will also evaluate the required training data, the model training time, and the time to transcribe actual data. | | |
| **Project Planning & Methodology** | | |
| **Project Planning:**  After the product has been researched and built upon, the model will be initially trained on a publicly available dataset. For languages like Nepali, Sinhala, Bengali and so on, OpenSLR provides high quality labelled data with manual quality check. The dataset consists of transcribed audio data for Nepali language in wav extension with TSV file containing audio ID/ File ID along with the transcription of the data. The project will use Wav2Vec2 algorithm which uses CNN with unsupervised pre-training and fine-tuning using labelled data. The model will be trained and fine-tuned using the OpenSLR dataset along with WER metrics calculation for training and testing. Then, the model will be tested on real world data with recorded audio or through microphone recorded data.  **Methodology:**  The project will follow the Agile Scrum methodology for project/ software development.  **Project Timeline:**    **Gantt Chart:**   |  |  |  |  | | --- | --- | --- | --- | | Task Name | Duration | Start | Finish | | **Final Project** | **56 days** | **Mon 4/26/21** | **Mon 7/12/21** | | **Initial Phase** | **9 days** | **Mon 4/26/21** | **Thu 5/6/21** | | Initial Topic Research | 4 days | **Mon 4/26/21** | Thu 4/29/21 | | Topic Feasibility Study | 2 days | Fri 4/30/21 | Mon 5/3/21 | | Project Goals and Methodology Study | 2 days | Tue 5/4/21 | Wed 5/5/21 | | **Planning Phase** | **3 days** | **Thu 5/6/21** | **Mon 5/10/21** | | **MS Project Creation** | **1 day** | **Thu 5/6/21** | **Thu 5/6/21** | | Make Gantt Chart | 1 day | Thu 5/6/21 | Thu 5/6/21 | | Requirement Analysis | 1 day | Fri 5/7/21 | Fri 5/7/21 | | Create Initial Project Plan | 1 day | Mon 5/10/21 | Mon 5/10/21 | | **Execution Phase** | **34 days** | **Tue 5/11/21** | **Fri 6/25/21** | | Initial Environment Setup | 2 days | Tue 5/11/21 | Wed 5/12/21 | | ASR Algorithm research | 10 days | Thu 5/13/21 | Wed 5/26/21 | | Create ASR engine in Python | 15 days | Thu 5/27/21 | Wed 6/16/21 | | Speech Data Collection | 4 days | Thu 6/17/21 | Tue 6/22/21 | | Model Training and Testing | 3 days | Wed 6/23/21 | Fri 6/25/21 | | **Monitoring and Controlling Phase** | **7 days** | **Mon 6/28/21** | **Tue 7/6/21** | | Performance Evaluation | 1 day | Mon 6/28/21 | Mon 6/28/21 | | Training Parameters Optimization | 3 days | Tue 6/29/21 | Thu 7/1/21 | | Re-Test Training/ Testing Performance | 3 days | Fri 7/2/21 | Tue 7/6/21 | | **Closing Phase** | **4 days** | **Wed 7/7/21** | **Mon 7/12/21** | | Setup Final Report and Submission | 4 days | Wed 7/7/21 | Mon 7/12/21 | | | |
| **Resources** | | |
| **The hardware and software I require to complete my Project successfully:**  Software:   1. Python 3.7 >= 2. Wav2Vec2 3. Hugging Face’s Implementation of Wav2Vec2 4. PyCharm or Python IDE 5. Flask   Hardware:   1. CUDA enabled graphics (for high performance in training) 2. PC with adequate RAM >=16GB, CPU and storage | | |
|  | |
| **Human Resource** | | |
| **I am working on my Project with the following people** | | |
| **Name:** M. Maharjan | **Role:**  Module Leader | |
|  |
| **Initial Bibliography** | | |
| * Gong, Y., Li, J., Deng, L. and Haeb-Umbach, R., 2016. *Robust automatic speech recognition - a bridge to practical applications*. Academic Press, pp.1-7. * O’Shaughnessy, D., 2008. Invited paper: Automatic speech recognition: History, methods, and challenges. *Pattern Recognition*, 41(10), pp.2965-2979. * Baevski et al. (2020). Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. <https://arxiv.org/abs/2006.11477> * Schneider et al. (2019). wav2vec: Unsupervised Pre-training for Speech Recognition. <https://arxiv.org/abs/1904.05862> * Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. 2018. [dataset] Directed by O. Kjartansson, S. Sarin, K. Pipatsrisawat, M. Jansche and L. Ha. Gurugram, India. | | |