



# Abhisek Panigrahi

## Curriculum Vitae

### Education

2012–2016 **B.Tech in Computer Science and Engineering**, *National Institute of Science and Technology*, Berhampur, 8.63(CGPA).

#### Projects:

- **Movie Recommendation System** : The system predicts rating by an user on a movie based on the user data and data of other users who are related to the target user - Implemented Restricted Boltzmann Machine, k-nearest neighbors. Read the dataset in the format of an user and a movie. Predicted rating by an user on a movie. Dataset: MovieLens 100k Dataset
- **Assembler using C** : The program generates machine language for IBM 360. Independently developed a C program which generates IBM 360 machine code for assembly language instructions load, store, add etc. Written two pass - define symbols, evaluate fields and generate code. Created symbol table, pseudo-op table, machine-op table. Processed the given assembly language instructions and converted into machine code using the tables.
- Presented a seminar on Amazon Echo

### Experience

Oct 2018 - **Specialist Programmer**, *Infosys*, Bangalore.  
Present

Feb 2017 - **Systems Engineer**, *Infosys*, Bangalore.  
Sep 2018

**MLOps** Description: setup MLOps for a project

Role: set up gitlab, gitlab runner, minio, docker registry. Created pipeline with 4 stages - create image and push to docker registry, push data, push model, deploy

**Abstract Summarization** Description: Summarize a paragraph

Role: Created torchserving format for `t5-small`, `sshleifer/distilbart-xsum-12-6` and `google/pegasus-xsum` models. Written client code to inference served models.

**Infosys HR model Creation** Description: Create Infosy HR model for Question Answering Task

|                                |   |
|--------------------------------|---|
|                                | <p><u>Role:</u> Fine tuned HR corpus on <i>distilbert-base-uncased</i> language model. Trained QA model on squad data using the resultant language model. Fine tuned squad format hr data on resultant QA model. Analyzed F1 on HR data after augmentation using paraphrasing and unwanted words removal. Reviewed 210 questions from HR data.</p>  |
| Question Generation            | <p><u>Description:</u> Generates questions given a paragraph, document or url</p> <p><u>Role:</u> Compared unilm and t5 model on Infosys policy document for question quality and inference time. Served t5 model using torchserve: torchserve image creation, archiving to torchservable format and serving. Added batch inference for torchservable format. <code>/paragraph</code> and <code>/doc</code> service endpoint creation. <code>/paragraph</code> - flask api endpoint to generate questions given a paragraph. <code>/doc</code> - flask api endpoint to generate questions given a document. Migrated flask to fastapi. Added aiohttp post request for async support. Added zip file support. Added containerization support with docker-compose. Recorded benchmark result for fastapi and flask.</p> |
| Reading Comprehension Platform | <p><u>Description:</u> Answer questions on paragraph or document</p> <p><u>Role:</u> Analyzed different datasets on BERT for correctness of answers and number of no answer. Recorded groundtruth answers for 200 questions on policy documents to compare different models. Added tokenization support for Bert, DistilBert, DistilRoberta. Analyzed answers from reading comprehension platform where input are questions from <i>Question generation</i> module, then augmented the questions using paraphrasing and analyzed answers again.</p>   |
| FAQ Extractor                  | <p><u>Description:</u> Learns similar questions from text dataset and gives closest match when a question is asked</p> <p><u>Role:</u> Contributed in model creation, word embedding creation, and initial model optimization. Served the model and <a href="#">stanford glove embedding</a> using tensorflow serving. Handled multiple service ids to support multiple dataset and multiple users. Added support to cache embedding and model in memory. Created deployment package using pyinstaller. Added containerization support using docker. Handled multiple service requests using celery</p> <p><u>tools:</u> tensorflow-serving, tensorflow, docker, pyinstaller, python language</p>   |
| Destination Assist             | <p><u>Description:</u> When user queries a destination, bot responds back with top 3 nearby points of interest including ratings and whether it is open or close.</p> <p><u>Role:</u> wrote handlers for the chatbot which includes maintaining the context throughout the conversation and switching the context for different kinds of query</p> <p><u>tools:</u> Java</p>  |
| Tasks helping team members     | <p><u>Description:</u> Small tasks helping team members</p> <p><u>Role:</u> Annotated 20 call transcripts on discovery, connection or objection. Compared 4 different models on question answering tasks as correctly answered or not.</p>  |
| Jun 2016 - Jan 2017            | <p><b>Systems Engineer Trainee, Infosys, Mysore.</b></p>  |

**Tennis** It is both command prompt and GUI based. It can: Search a player in the database, Update or Add Player, Book a match, View or Cancel a match, View tournament details

tools: tkinter, cx\_Oracle, python language

---

## Certificate and Recognition

- [Reinforcement Learning](#)
- [Deep Learning Specialization](#)

August 2018 Most Valuable Player Award

VP - Head Infosys Center of Emerging Technology Solutions

---

## Paper Reimplementation

[AlexNet](#)

[paper](#)

[VGG](#)

[paper](#)

---

## Technical Skills

tensorflow

keras

python advanced

git

c++

docker

java

ubuntu

C