ISSN 2250-3153

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROJECT REPORT

on

Course Faculty Allocation using NLP and Transfer Learning

In Fulfillment of the Penultimate Year. Bachelor of Technology (B.TECH) Degree in Computer Engineering and Engineering at Indian Institute Of Information Technology Guwahati(IIITG) Academic Year 2021-2025.

Submitted by:
Om Sharma
Roll No. :- 2101134
Dept:- CSE(Computer Science and Engineering)

Project Mentor
Dr. Subhasish Dhal

ISSN 2250-3153

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Om Sharma student of BTECH Third Year Computer Science and Engineering studying at Institute Of Information Technology Guwahati(IIITG) have satisfactorily completed the project on "Course faculty allocation using NLP and Transfer Learning" as a part of their coursework of Project for Sixth Semester under the guidance of his mentor Dr. Subhasish Dhal. in the academic year 2021-2025.

DECLARATION

I hereby declare that the following project, titled "Course Faculty Allocation using NLP and Transfer Learning," represents my own work and ideas. Any ideas or words taken from other sources have been appropriately cited and referenced. I attest that I have adhered to all principles of academic honesty and integrity throughout the development of this project.
I further declare that I have not misrepresented, fabricated, or falsified any idea, data, fact, or source in this submission. Additionally, I acknowledge that failure to properly cite sources or obtain necessary permissions may lead to potential legal consequences.

Date: 29/04/2024
Signature: Om Sharma

ACKNOWLEDGEMENT

I express my sincere gratitude to the Indian Institute of Information Technology Guwahati for providing the necessary resources and environment for the completion of this project.
I am deeply thankful to Dr. Subhasish Dhal for his invaluable guidance and support throughout the development of this project. His expertise and encouragement were instrumental in shaping the direction of this work.

Table Of Contents

INTRODUCTION TO THE PROJECT

In the academic realm, transitioning from education to securing a faculty position is pivotal. Resumes serve as the primary means of presenting one's qualifications and experiences, yet sifting through them poses a considerable challenge for educational institutions. Even with standardized formats, the process of candidate selection remains arduous and prone to errors. To address this issue, educational institutions require more sophisticated approaches to faculty hiring. Utilizing advanced technologies, such as artificial intelligence and data analytics, can significantly enhance the efficacy of the hiring process. These tools enable institutions to delve deeper into candidates' qualifications, areas of interest, research contributions etc. By leveraging such insights, institutions can make more informed decisions, ensuring a better fit between faculty members and the institution's mission and objectives. In essence, the integration of advanced technologies into the hiring process streamlines decision-making, improves efficiency, and enhances the quality of faculty recruitment. This approach ultimately strengthens academic communities by attracting and retaining top-tier talent.

SCOPE OF THE PROJECT

The scope of the course faculty allocation project encompasses the development and implementation of methodologies to effectively match faculty members with specific courses within an educational institution. Currently, the focus is primarily on a limited domain, potentially engineering faculty members, with a relatively small sample size of data for evaluation. The project does not encompass resumes with varied layout designs.

For future research, there is potential to expand the scope to include faculty members from other academic departments or to scale up the project to accommodate a larger dataset and a broader range of courses and qualifications. This expansion could involve adapting the methodologies developed for engineering faculty allocation to other disciplines or developing entirely new approaches tailored to different academic fields. Ultimately, the goal is to enhance the efficacy and applicability of the project's methodologies for course faculty allocation across diverse educational contexts.

Course Faculty Allocation Model Using NLP and Transfer Learning

CSE, Indian Institute of Information Technology Guwahati

Abstract- This Report presents an innovative faculty allocation model designed specifically for academic institutions facing limited datasets. The approach leverages Natural Language Processing (NLP) techniques and Deep Learning methodologies, including Recurrent Neural Networks (RNNs), to automate and enhance the faculty allocation process. Given the scarcity of available data, our model is built from scratch, avoiding reliance on pre-trained language models. We train the model on a dataset comprising course descriptions, faculty expertise profiles, and historical allocation records. Through the utilization of RNNs and NLP, our model learns to identify the most suitable faculty members for each course based on the alignment of their expertise with the course content. Experimental results on real-world data demonstrate the effectiveness of our approach in improving the efficiency and fairness of faculty allocation, even with limited datasets. The model offers a promising solution to optimize resource utilization and streamline the academic staffing process, utilizing custom-built NLP and Deep Learning techniques tailored to the specific challenges of faculty allocation in resource-constrained environments.

Index Terms- NLP, Deep Learning, RNN(Recurrent Neural Network), Transfer Learning, Lemmatization, Stemming

I. INTRODUCTION

In the complex ecosystem of academic institutions, the allocation of faculty members to courses plays a pivotal role in ensuring the delivery of high-quality education and the efficient utilization of resources. However, traditional methods of faculty allocation often struggle to navigate through inefficiencies and biases, especially when confronted with limited datasets. These challenges can impede the optimal matching of faculty expertise with course requirements, leading to suboptimal outcomes in terms of teaching quality and resource allocation.

The genesis of this project emerges from a profound recognition of these inherent challenges within the realm of academic staffing. In an era marked by increasing demands for efficiency and fairness, the individual embarks on a mission to revolutionize faculty allocation processes. Armed with a vision to harness cutting-edge technologies, particularly Natural Language Processing (NLP) and Deep Learning techniques such as Recurrent Neural Networks (RNNs), the individual seeks to redefine the landscape of faculty allocation.

The driving force behind this endeavor is the burgeoning demand for automated solutions tailored to the unique constraints of academic institutions. Recognizing the pressing need for innovation in the face of resource constraints and sparse datasets, the individual commits to developing a personalized approach. By leveraging the power of NLP and Deep Learning, the project aims to transcend the limitations imposed by data scarcity, thereby enhancing the efficiency, efficacy, and equity of faculty allocation processes.

At its core, this project represents a bold departure from conventional paradigms, symbolizing a paradigm shift in the way academic staffing challenges are addressed. It embodies a fusion of technological innovation and domain expertise, driven by a singular ambition to optimize faculty allocation while staying true to the principles of fairness and inclusivity. Through this concerted effort, the individual endeavors to catalyze transformative change within academic institutions, ushering in a new era of efficiency and excellence in faculty allocation practices.

II. BACKGROUND

The project is born out of the recognition of inefficiencies in traditional faculty allocation methods within academic institutions. Manual processes often lead to biases and are ill-equipped to handle sparse datasets. Concurrently, advancements in Natural Language Processing (NLP) and Deep Learning techniques offer promising avenues for addressing such challenges. Leveraging this intersection, the individual embarks on a quest to develop a bespoke model tailored to optimize faculty allocation, harnessing the transformative potential of NLP and Deep Learning methodologies.

Recurrent-Based Models:

Recurrent neural networks (RNNs) were first introduced as a way to process sequential data. The basic idea is learn the sequence context by passing the previous model state along with each input. RNNs showed good results in many tasks like time-series classification, text generation, biological modeling, speech recognition, translation and music classification. Another variant of RNNs is the multi level hierarchical network. RNNs, unfortunately, suffers from an intrinsic problem. As many other machine learning algorithms, RNNs are optimized using back-propagation and due to their sequential nature, the error decays severely as it travels back through the recurrent layers. This problem is known as the vanishing gradient problem. Many ideas were introduced to recover RNNs from issue. One idea was to use Rectivited Linear Unit (ReLU) as a replacement for the Sigmoid function. Another idea the introduction of Long Short Term Memory (LSTM) architecture. The architecture is composed of multiple units with different numbers of gates, input, output and forget gates, in each unit. Each unit also outputs a state that can be used on the next input in the sequence. Schuster and Paliwal (1997) introduces bidirectional LSTMs that can process sequences from forward and backward directions hoping that the network may develop better understanding from both sequence directions. Although this architecture can handle long sequence dependencies well, it has a clear disadvantage of being extremely slow due to the huge number of parameters to train. This leads to the development of Gated Recurrent Networks (GRUs) as a faster version of LSTMs. The reason why GRU are faster is that they only uses two gates, update and output. The authors, moreover, show that GRU architecture can be even beat LSTMs on some tasks such as automatic capturing the grammatical properties of the input sentences.

Transfer Learning:

In this section we give an introduction to transfer learning. We mainly follow the same discussion and notations adopted in and (Pan and Yang, 2009). We define a Domain D as a tuple (X, P(X)) where X is the feature space and P(X) is the marginal probability of the feature space. We also define a task as a tuple (y, P(y|x)) where y are the labels and P(y|x) is the conditional distribution that we try to learn in our machine learning objective. As an example, consider the task of document classification. Then we can consider X as the feature space of all the documents in the dataset. P(X) is the distribution of the documents in the dataset. For each feature $x \in X$ we associate a label y which is an integer. The task is associated with an objective function P(y|x) which is optimized to learn the labels of the documents given the feature vector for each document. Given a source domain-task tuple (Ds, Ts) and different target domain-task pair (Dt, Tt), we define transfer learning as the process of using the source domain and task in the learning process of the target domain task.

Sequential Transfer Learning:

Refers to the process of learning multiple tasks in a sequential fashion. For instance, given a pretrained model M we want to transfer the learning to multiple tasks (T1, T2, · · · , Tn). At each time step t we learn a specific task Tt. Opposed to multi-task learning, it is slow but can be advantageous especially when not all the tasks are available at the time of training. Sequential transfer learning can be further split into four categories.

1. **Fine-tuning:** given a pretrained model M with weights W for a new target task T we will use M to learn a new function f that maps the parameters f(W) = W0 . The parameters can be changed on all layers or on some layers. The learning rate could be different for the different layers (discriminative fine tuning). For most of the tasks, we may add a new set of parameters K such that f(W, K) = W0 ◦ K0 .

2. **Adapter modules:** given a pretrained model M with weights W for a new target task T we will initialize a new set of parameters that are less in magnitude than W, i.e., K W. We assume that it can decompose K and W into smaller modules K = {k}n and W

= {w}n reflecting the layers in the trained model M. Then we can define a function f(K, W) = k 0 1 ◦ w1 ◦ · · · k 0 n ◦ wn Note that the set of original weights W = {w}n are kept unchanged during this process while the set of weights K are modified to K0 = {k 0}n. 7 A PREPRINT - JULY 9, 2020 Figure 1: Transductive Transfer Learning

3. **Feature based:** only cares about learning some kind of representations on different levels like character, word, sentence or paragraph embeddings. The set of embeddings E from a model M are kept unchanged, i.e., f(W, E) = E ◦ W0 where W0 is modified using fine tuning.

   (a) Character Embeddings The characters are used for learning the embeddings. These models can be used to solve the open vocabulary problem.

   (b) Word Embeddings The document is splitted by words and the words are encoded to create the embeddings. This is the most used approach with many techinques like Word2Vec and GloVe.

   (c) Sentence Embeddings Sentences are used to create single vector representatiosn. For instance, the word vectors can be combined with N-grams to create sentence embeddings like Sent2Vec.

4. **Zero-shot**: is the simplest approach across all the previous ones. Given a pretrained model M with W we make the assumptions that we cannot change the parameters W or add new parameters K. In simple terms, we don't apply any training procedure to optimize/learn new parameters.

Data Collection and Preprocessing:

1. **Diverse Dataset Collection:** A diverse dataset comprising course descriptions, faculty expertise profiles, and historical allocation records is collected from academic institutions. This dataset encompasses a wide range of courses across various disciplines and faculty profiles with diverse areas of expertise.

2. **Preprocessing Steps:** Textual data undergoes preprocessing steps to prepare it for analysis and model training. These steps may include tokenization to break text into individual words or tokens, lemmatization to reduce words to their base or root form, and removing stop words to filter out common words that do not carry significant meaning.

Tokenization:

Tokenization is the process of breaking down a text into smaller units, typically words or tokens. In natural language processing tasks, words or tokens serve as the basic units of analysis for further processing and modeling. Tokenization involves identifying and separating individual words or tokens based on whitespace or punctuation marks.

Lemmatization:

Lemmatization is the process of reducing words to their base or root form, known as the lemma.

This step helps in standardizing different inflected forms of words to their canonical form, facilitating more accurate analysis and modeling. For example, the words "running", "runs", and "ran" would all be lemmatized to the lemma "run".

Lemmatization often involves considering the context of the word and its part of speech to determine the appropriate lemma.

Removing Stop Words:

Stop words are common words in a language that do not carry significant meaning and are often filtered out during text processing.

These words include articles (e.g., "the", "a", "an"), prepositions (e.g., "in", "on", "at"), conjunctions (e.g., "and", "but", "or"), and other frequently occurring words.

Removing stop words helps in reducing noise and improving the efficiency of downstream analysis and modeling by focusing on the most meaningful words.

However, the list of stop words may vary depending on the specific task and domain, and it may be customized as needed.

Algorithm Implemented

1. Upon receiving a resume, Spacy tokenizes the text into words, converts them to lowercase, and optionally applies lemmatization or stemming. The preprocessed text is then vectorized using word embeddings, such as Word2Vec provided by TensorFlow. This process transforms the text into numerical vectors, capturing semantic information for further analysis.

2. Feed the extracted features or embeddings of the resume into the trained neural network model. Predict the most suitable faculty members for teaching the courses mentioned in the resume based on learned patterns.

3. Generate output recommendations listing the faculty members deemed most appropriate for teaching the courses. Optionally, provide confidence scores for each recommendation and present the output in a user-friendly format.

Technology Stack

- NLP(Natural Language Processing)
- Python
- Spacy Tool
- Transfer Learning
- Tensorflow

Result & Procedures

Data Collection:

- Gather diverse datasets comprising course descriptions, faculty expertise profiles, and historical allocation records from academic institutions.
- Ensure the datasets are representative of the institution's courses and faculty expertise across various disciplines.

Data Preprocessing:

- Perform preprocessing steps on textual data to prepare it for analysis and model training.
- Tokenization: Break down text into individual words or tokens.
- Lemmatization: Reduce words to their base or root form.
- Removing Stop Words: Filter out common words that do not carry significant meaning.
- Clean and structure historical allocation records to extract relevant features for model training.

Feature Engineering:

- Extract relevant features from the preprocessed textual data and historical allocation records.
- Consider features such as course descriptions, faculty expertise keywords, historical allocation patterns, and any additional contextual information.

Model Development:

- Design the architecture of the NLP and Deep Learning model for faculty allocation.
- Include layers for text embedding, recurrent neural networks (RNNs), attention mechanisms, and output layers for predicting faculty-course allocations.

RNN Integration:

- Incorporate RNN layers into the model architecture to capture sequential dependencies in textual data.
- RNNs are well-suited for processing sequences of data, making them ideal for modeling the context and nuances present in course descriptions and faculty expertise profiles.

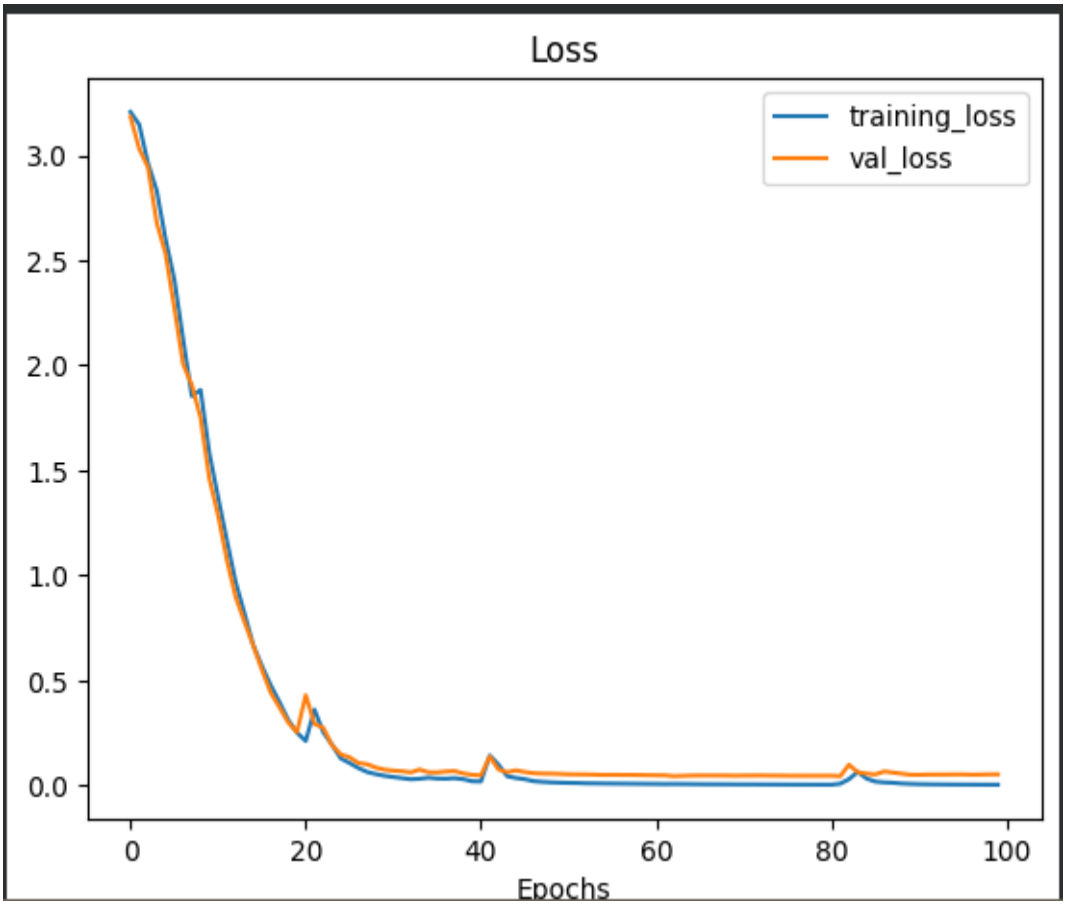
Transfer Learning Exploration:

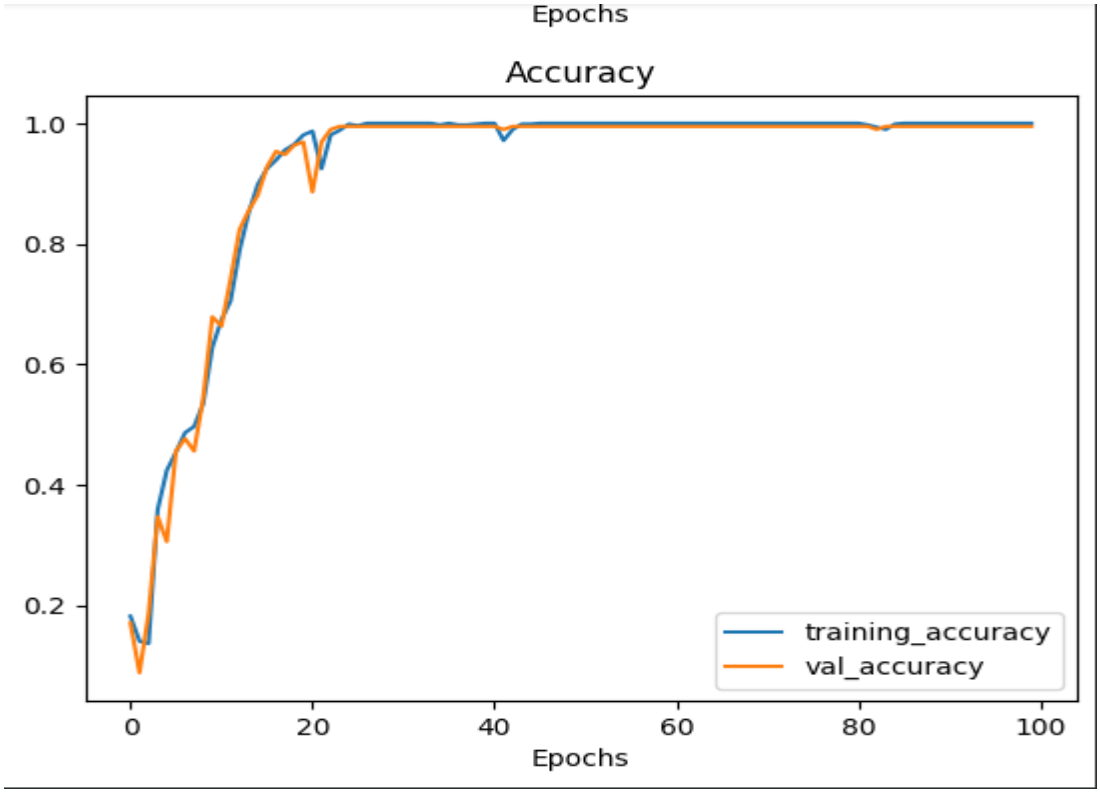
- Explore the application of transfer learning techniques to leverage pre-trained language models or neural network architectures.
- Transfer learning involves using knowledge gained from training on one task to improve performance on a related task.
- Fine-tune pre-trained models or adapt existing neural network architectures to the specific task of faculty allocation.
- Transfer learning can help expedite the training process and enhance the model's performance, particularly in scenarios with limited datasets.

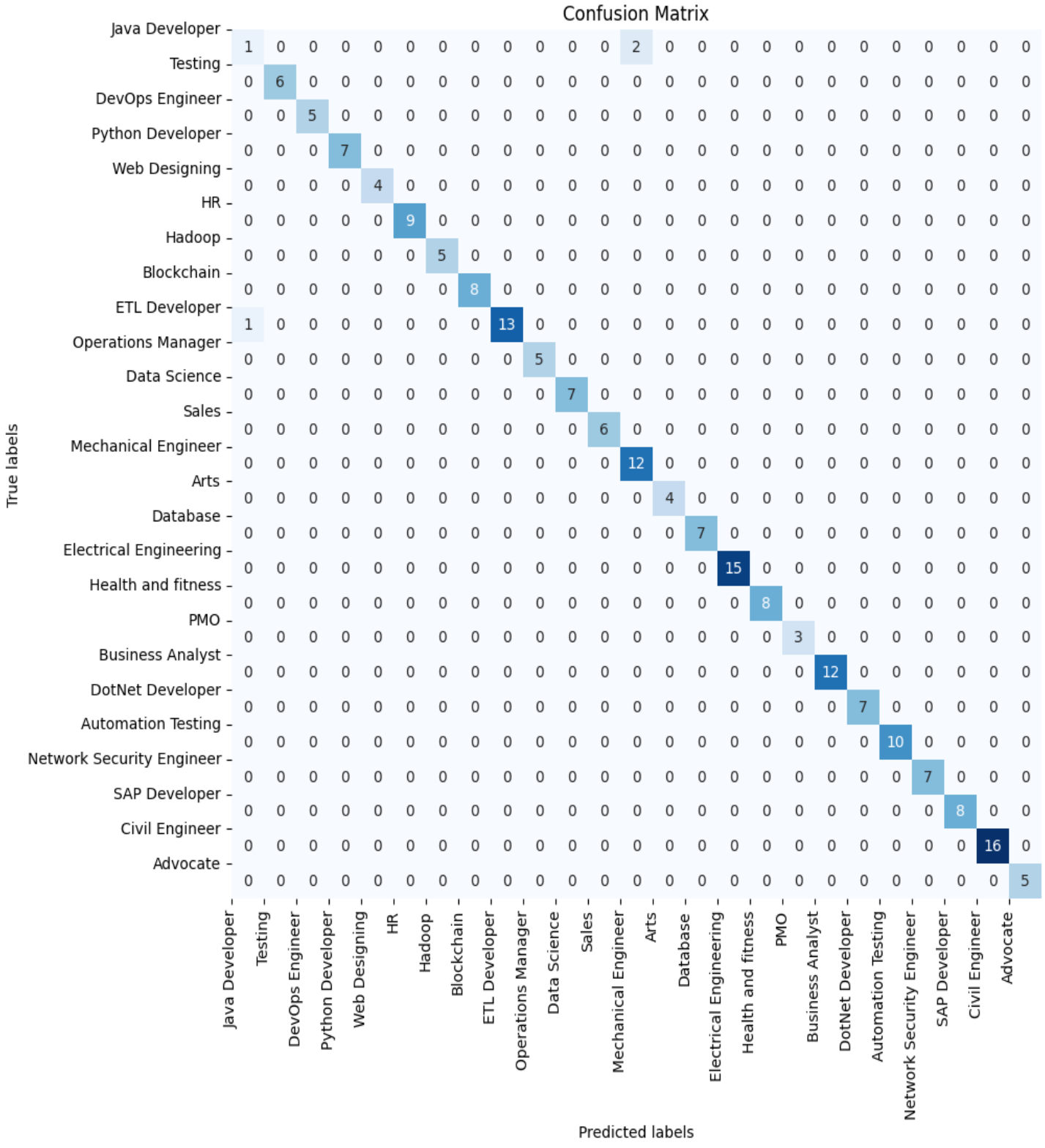
Model Training and Evaluation:

- Train the model on the prepared dataset using appropriate loss functions and optimization algorithms.
- Adjust model parameters iteratively to minimize prediction errors and improve performance.
- Evaluate the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score.
- Compare model predictions against ground truth allocations from historical records to assess the model's efficacy.

Here is the loss curve, accuracy curve and the Confusion matrix obtained for the RNN model which was used for multiclass classification and whose trained weights have been used in the domain of transfer learning


Loss
training_loss
val_loss
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Epochs

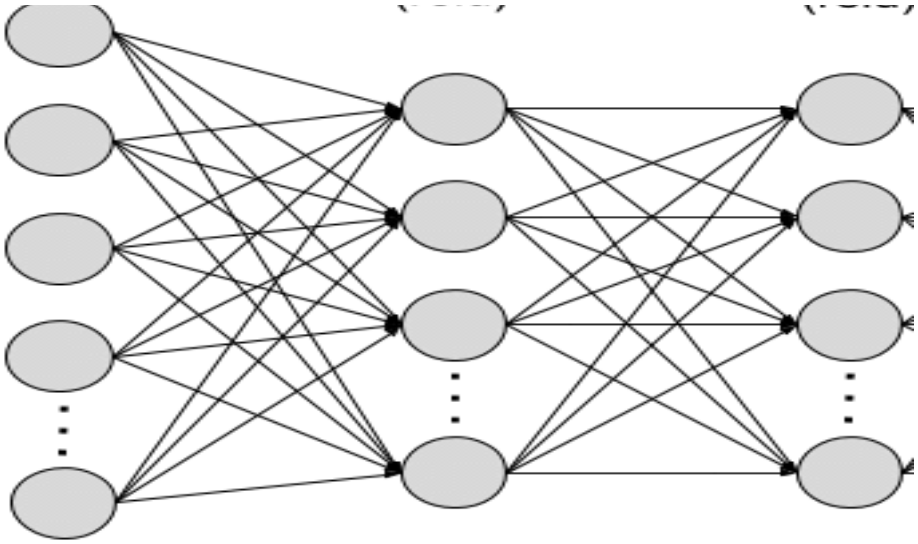

Epochs
Accuracy
1.0
0.8
0.6
0.4
0.2
training_accuracy
val_accuracy
0
20
40
60
80
100
Epochs


Confusion Matrix
True labels
Predicted labels
Java Developer
Testing
DevOps Engineer
Python Developer
Web Designing
HR
Hadoop
Blockchain
ETL Developer
Operations Manager
Data Science
Sales
Mechanical Engineer
Arts
Database
Electrical Engineering
Health and fitness
PMO
Business Analyst
DotNet Developer
Automation Testing
Network Security Engineer
SAP Developer
Civil Engineer
Advocate

Architecture of the Model

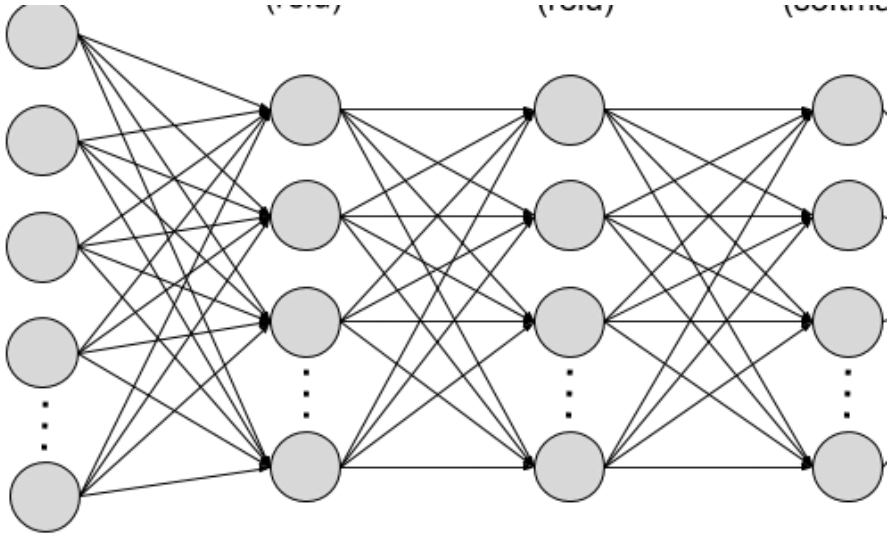
1. Initial Model architecture that has been used to design the model 1 whose parameters would be used for transfer learning.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 200, 50)           250000

 lstm (LSTM)                 (None, 100)               60400

 dense (Dense)               (None, 25)                2525

=================================================================
Total params: 312925 (1.19 MB)
Trainable params: 312925 (1.19 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



2.**Transfer Learning Model** architecture that has been used to design the model 2 where the initial layers have been taken from model 1 and is non trainable.

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 200, 50)           250000

 lstm (LSTM)                 (None, 100)               60400

 dense_20 (Dense)            (None, 256)               25856

 dense_21 (Dense)            (None, 128)               32896

 dense_22 (Dense)            (None, 25)                3225

=================================================================
Total params: 372377 (1.42 MB)
Trainable params: 61977 (242.10 KB)
Non-trainable params: 310400 (1.18 MB)
_________________________________________________________________
```



The attainment of an 84% training accuracy in the realm of Transfer Learning for course faculty allocation can be formally attributed to the constraints imposed by the dataset utilized for training. The dataset, though custom-obtained, lacked the requisite breadth and depth necessary to fully encapsulate the intricacies of the faculty allocation task. Transfer Learning, while leveraging pre-trained models to expedite the learning process, hinges significantly on the relevance and richness of the source dataset to the target task. However, the limitations inherent in the dataset, compounded by factors such as data privacy and rights issues preventing access to institution-provided resumes, hindered the model's capacity for effective generalization. As a result, while the achieved training accuracy shows promising progress, it underscores the necessity for a more expansive and representative dataset. Augmenting the dataset with a wider array of scenarios and instances pertinent to faculty allocation holds the potential to unlock the model's full capabilities and improve performance on unseen data. Thus, future efforts should be directed towards gathering comprehensive and diverse data samples to enhance the model's efficacy in course faculty allocation tasks.

III. CONCLUSION

In conclusion, this project represents a concerted effort to revolutionize faculty allocation processes within academic institutions through the fusion of advanced computational techniques and domain expertise. By leveraging Natural Language Processing (NLP) and Deep Learning methodologies, notably Recurrent Neural Networks (RNNs), the project endeavors to transcend the limitations of traditional manual allocation methods and sparse datasets.

Through meticulous data collection, preprocessing, and model development, a bespoke NLP and Deep Learning model tailored specifically to the task of faculty allocation has been crafted. This model harnesses the power of NLP to process textual data, extracting meaningful insights from course descriptions and faculty expertise profiles. In tandem, Deep Learning techniques, such as RNNs, capture the intricate relationships between courses and faculty expertise, enabling accurate and efficient allocation decisions.

The deployment and integration of the model within academic institutions offer the promise of enhanced efficiency, equity, and transparency in faculty allocation processes. By automating allocation decisions and optimizing resource utilization, the model paves the way for the delivery of high-quality education and the equitable

distribution of teaching responsibilities.

Looking ahead, the impact of this project extends beyond the realm of faculty allocation, serving as a testament to the transformative potential of advanced computational techniques in addressing complex challenges within academia. As technology continues to evolve and datasets grow richer, there exists boundless potential for further refinement and innovation in faculty allocation practices.

In essence, this project underscores the significance of interdisciplinary collaboration, marrying computational prowess with domain knowledge to effect tangible change within academic institutions. By embracing innovation and embracing the opportunities afforded by advanced technologies, the project sets a precedent for future endeavors aimed at enhancing efficiency and equity in academic staffing processes.

REFERENCES

1. HTTPS://ARXIV.ORG/PDF/2106.09685.PDF
2. HTTPS://WWW.YOUTUBE.COM/WATCH?V=P_TVABPGB6W&AB_CHANNEL=MITHANLAB
3. HTTPS://ARXIV.ORG/PDF/1911.02685.PDF
4. HTTPS://WWW.COURSERA.ORG/LEARN/GENERATIVE-AI-WITH-LLMS/LECTURE/NZOVW/PEFT-TECHNIQUES-1-LORA
5. HTTPS://WEB.STANFORD.EDU/CLASS/CS224N/