DOCUMENT TRANSLATION

MODEL:

1. Introduction:

The aim of this project is to develop a system capable of recognizing text formatting styles in pdf documents using its LaTeX formatting and while preserving those formats to convert to other language .LaTeX is a widely used document preparation system for typesetting technical and scientific documents. Identifying text formatting styles such as bold, italic, and underline along with its size

2. Problem Statement:

Create a simple yet effective model for document translation capable of converting text from one language to another while preserving the document's structure and formatting. The model should handle various document types and languages, prioritizing accuracy, efficiency, and ease of integration into existing document processing pipelines.

3. Methodology:

The project workflow involves several key steps:

A. LaTeX Document Processing:

This was used to give labels for each document based on font style for each word and font size which is present in the LaTeX script.

B. PDF to Image Conversion:

PDF documents are converted to images to facilitate optical character recognition (OCR) using the PyMuPDF library.

C. Optical Character Recognition (OCR):

OCR is performed on the images using the Keras-OCR library, which utilizes a convolutional neural network (CNN) for text detection and recognition. Detected words along with their bounding boxes are obtained.

D. Feature Extraction:

Features are extracted from the detected words using a pre-trained VGG16 model, a convolutional neural network (CNN) originally trained for image classification tasks. These features capture the visual characteristics of the text.

E. Data Labeling:

Extracted words are labeled based on their formatting styles (bold, italic, underline) obtained from the LaTeX document processing step.

F. Machine Learning Classification:

Extracted features and labels are used to train machine learning classifiers including Random Forest and Support Vector Machine (SVM) for text style classification.

G. Evaluation:

The performance of the classifiers is evaluated using metrics such as accuracy score and classification report to assess the effectiveness of the proposed approach.

H.Regression Model for Font Size:

A basic CNN regression model using features extracted from the VGG16 architecture. The model takes in features extracted from images using VGG16 and learns to predict font sizes for each word based on these features.

4. Results and Discussion:

The machine learning classifiers achieved promising results in recognizing text formatting styles in LaTeX documents.Random Forest and SVM classifiers demonstrated high accuracy in classifying text styles, indicating the effectiveness of the feature extraction and classification approach.Challenges such as handling complex LaTeX syntax, variations in formatting styles, and noisy OCR outputs were addressed through careful preprocessing and feature engineering.The system's ability to accurately classify text formatting styles has potential applications in document analysis, information retrieval, and content understanding tasks.

5. Conclusion:

The developed system provides an automated solution for recognizing text formatting styles in LaTeX documents.
By combining OCR with machine learning techniques, accurate classification of text styles is achieved, enhancing the usability of LaTeX documents in various applications.
Future work may focus on improving robustness to handle a wider range of LaTeX document styles, optimizing feature extraction techniques, and exploring advanced machine learning algorithms for text classification tasks.

Automation Pipeline for LaTeX Document Processing

Introduction:

In the modern era of digital documentation, the processing of LaTeX documents is essential for various academic, scientific, and professional purposes. LaTeX provides a powerful typesetting system that is widely used for producing documents with complex layouts, such as academic papers, reports, and presentations. However, processing LaTeX documents manually can be time-consuming and error-prone. To address this challenge, an automation pipeline has been developed to streamline the processing of LaTeX documents, including text extraction, compilation, and content analysis.

1. Text Extraction from PDF:
The first step in the automation pipeline is to extract text from PDF documents. PDF files are a common format for storing LaTeX documents, but directly processing PDFs can be challenging due to their complex structure. To overcome this challenge, the PyPDF2 library is used to extract text from PDF files. The extract_text_from_pdf function reads a PDF file and extracts text from each page using PyPDF2. This extracted text serves as the input for further processing in the pipeline.

2. LaTeX Compilation:
Once the text is extracted from the PDF, the next step is to compile LaTeX documents. LaTeX documents are written in a markup language that requires compilation to generate the final output in PDF format. The compile_latex function automates this compilation process using the latexmk utility. This function takes a LaTeX file as input, compiles it into a PDF, and saves the resulting PDF in a specified output directory. Additionally, it handles errors and exceptions that may occur during compilation, ensuring a smooth processing workflow.

3. Random Word Generation:
Random word generation is a useful feature in document processing for various purposes, such as testing, data augmentation, and content generation. In this pipeline, random words are generated using the random and string modules in Python. These random words can be used to augment text data or generate synthetic documents for testing and evaluation purposes.

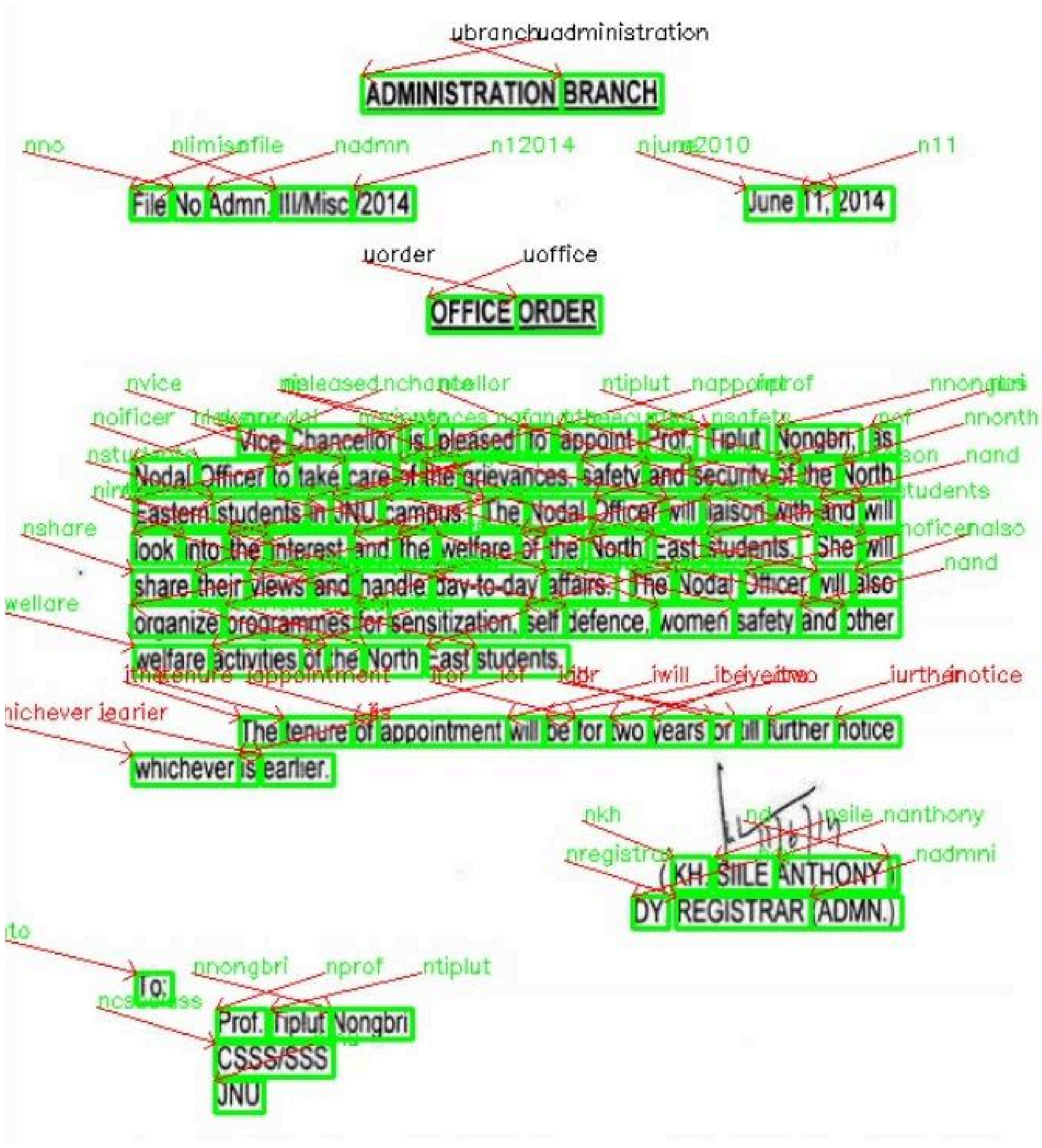
4. LaTeX Script Generation:
The final step in the automation pipeline is the generation of LaTeX scripts. These scripts can be used to create LaTeX documents with specific formatting styles, such as bold, italic, underline, and regular text. The latex_script function generates LaTeX

scripts by randomly selecting words from a text file and dividing them into four parts: regular, bold, italic, and underline. These scripts provide a customizable template for generating LaTeX documents with diverse content and formatting styles.

Conclusion:

The automation pipeline described in this report streamlines the processing of LaTeX documents by automating text extraction, LaTeX compilation, random word generation, and LaTeX script generation.

MAKING BOUNDING BOXES


ubranchuadministration
nno
nadmn
n12014
n11
uorder
uoffice
nvice
noificer
nshare
wellare
nnonth
nand
nand
nkh
nanthony
nadmni
nregistrar
ito
nnongbri
nprof
ntiplut

ADMINISTRATION BRANCH

File No Admn. III/Misc/2014

June 11, 2014

OFFICE ORDER

Vice Chancellor is pleased to appoint Prof. Tiplut Nongbri, as Nodal Officer to take care of the grievances, safety and security of the North Eastern students in JNU campus. The Nodal Officer will liaison with and will look into the interest and the welfare of the North East students. She will share their views and handle day-to-day affairs. The Nodal Officer will also organize programmes for sensitization, self defence, women safety and other welfare activities of the North East students.

The tenure of appointment will be for two years or till further notice whichever is earlier.

[illegible]

(KH. SIILE ANTHONY)
DY. REGISTRAR (ADMN.)

To:
Prof. Tiplut Nongbri
CSSS/SSS
JNU

REGRESSION MODEL TO PREDICT FONT SIZE

Introduction:

- **Objective:** Develop a regression model to predict font sizes in LaTeX documents.
- **Source:** LaTeX files paired with corresponding PDF documents.
- **Features:** Textual features (word frequencies) and image features (extracted from PDF pages).
- **Target Variable:** Font size of text segments within LaTeX documents.
- **Data Preprocessing:** Parsing LaTeX files, extracting text content, converting PDF pages to images, and feature extraction using pre-trained models.

Exploratory Data Analysis (EDA):

- Summary statistics and visualization of font size distribution.
- Correlation analysis between font size and other features.

Model Selection:

Regression chosen for its ability to predict continuous numeric values.

Feature Engineering:

- Textual features derived from word frequencies and patterns in LaTeX files.
- Image features obtained by passing PDF page images through a pre-trained CNN model.

Model Working:

- Regression model takes combined textual and image features as input.
- Multiple layers of neurons perform nonlinear transformations on input data.
- Model learns to minimize the difference between predicted and actual font sizes through parameter adjustments during training.

Model Training:

- Trained using subset of dataset, with features as input and font sizes as target variable.

- Gradient descent optimization adjusts model parameters to minimize loss function.
- Tuned training parameters include learning rate, batch size, and number of epochs.

Model Evaluation:

- Evaluated on separate test dataset using metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE).
- Visualization of predicted font sizes versus actual font sizes provides insights into model performance.

Conclusion:

Regression model demonstrates promising performance in predicting font sizes based on textual and image features.Further refinement and optimization may enhance model performance for real-world applications.

Feature Extraction by Using Bounding Boxes

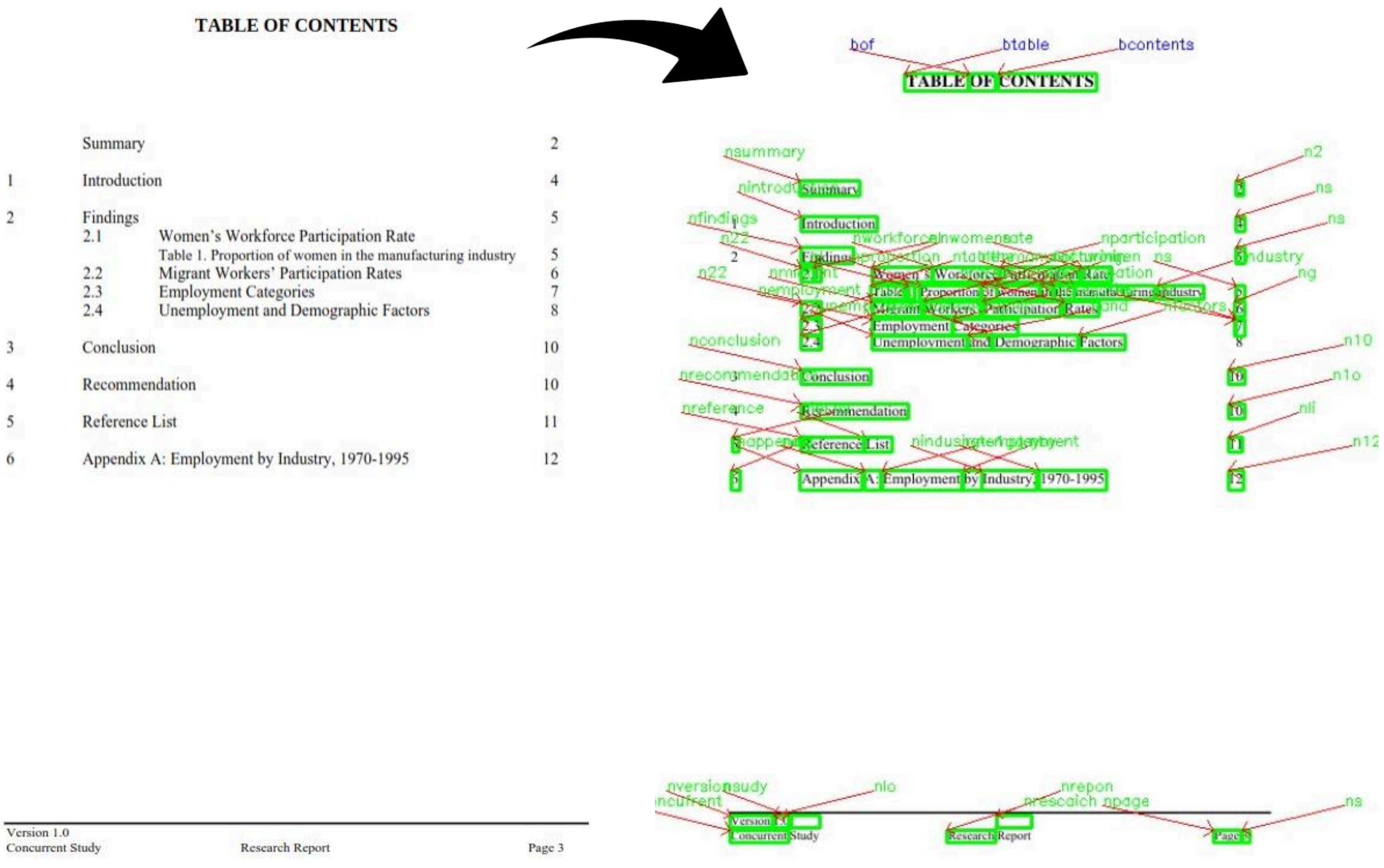


TABLE OF CONTENTS

	Summary		2
1	Introduction		4
2	Findings		5
	2.1	Women's Workforce Participation Rate	5
		Table 1. Proportion of women in the manufacturing industry	5
	2.2	Migrant Workers' Participation Rates	6
	2.3	Employment Categories	7
	2.4	Unemployment and Demographic Factors	8
3	Conclusion		10
4	Recommendation		10
5	Reference List		11
6	Appendix A: Employment by Industry, 1970-1995		12

Version 1.0
Concurrent Study Research Report Page 3

PDF Text Extraction Function: The code begins with a function designed to extract text from PDF files. It employs the PyPDF2 library, a popular tool for working with PDF files in Python. This function iterates through each page of the PDF, aiming to extract text.

Compilation of LaTeX Documents: The script provides two versions of the compile_latex() function, both serving the purpose of compiling LaTeX documents. The primary functionality involves utilizing latex, a tool widely used for automating the process of generating LaTeX documents. The code ensures error handling by checking for the existence of the LaTeX file and providing informative error messages when encountering issues during compilation. The second version of the function introduces additional functionality by moving the resulting PDF to a specified output directory. This enhancement facilitates better organization of compiled documents. Overall, the compilation process is robust and efficiently managed, with appropriate error handling mechanisms in place.

Generation of LaTeX Scripts: The section responsible for generating LaTeX scripts demonstrates a process of constructing formatted text sections within LaTeX documents. These scripts are generated programmatically, with each section containing regular, bold, italic, and underlined text. The source text for these sections is randomly selected from a provided text file (bulk4.txt). The script divides the selected words into four categories and formats them according to LaTeX syntax. However, the purpose or context of this script generation is not explicitly stated within the code. Adding comments or documentation to clarify the intention behind generating these LaTeX scripts would enhance the code's comprehensibility.

Random Word Generation: The code includes a segment dedicated to generating random words. It utilizes lowercase alphabets to generate words of varying lengths, creating a list of 100 random words. While this functionality showcases the use of Python's random and string modules, its relevance to the overall script is not apparent. Including a brief explanation or context for this random word generation would improve understanding, particularly regarding its connection to the broader script's objectives.

Introduction to Pipeline(How they all connected):

The code presented implements a comprehensive system for detecting text in document images, translating it into another language, and overlaying the translated text onto the original image. This system is designed to address the increasing demand for automated methods to extract and translate text from images in various languages. By combining optical character recognition (OCR), machine learning models, and natural language processing (NLP) techniques, the system aims to provide a versatile solution for document analysis and understanding.

System Architecture:

The proposed system consists of several interconnected modules, each serving a specific function in the text extraction, translation, and annotation pipeline. These modules include:

- Defines a class Test_Data to encapsulate functionalities related to processing test data. Initializes attributes such as PDF path, image path, features array, words list, and word-box list.
- Implements a method convert_to_img to convert PDF pages into image files. Utilizes the fitz library to open the PDF, extract pages, and save them as PNG images.
- A method features_data to extract features from the detected words in the images using the **VGG16 model**. Utilizes the keras_ocr pipeline for text recognition and the VGG16 model for feature extraction.
- Implements a method prediction to make predictions using trained models and extracted features. Utilizes **SVM** and font size prediction models to predict font styles and sizes for the detected words.
- Loads pre-trained SVM and font size prediction models. Initializes a Test_Data object with the PDF path and performs predictions on the test data.
- Constructs a merged list containing words, their bounding boxes, labels, and font sizes. Draws bounding boxes around detected words and overlays arrows and text annotations based on their font styles onto the image.
- Utilizes the **Google Translate API** to translate each word in the merged list to any language and prints the translated text.
- A inpaint_text utility functions for calculating midpoint coordinates and inpainting text regions in the image using the OpenCV library. Applies the inpainting function to remove text regions from the image.
-
    - **Translator Initialization**: The translator object is initialized to handle text translation using the Googletrans library.

- **Image Processing Loop**: A loop iterates over each word in the merged_list, adjusting bounding box coordinates and translating text for rendering.
- **Bounding Box Adjustment**: Conditions are applied to adjust bounding box coordinates, ensuring uniform placement of translated text.
- **Text Translation**: Each word is translated into Hindi using the translator.translate() method.
- **Font and Text Rendering**: Based on the word's label, appropriate font paths are selected, and text is rendered onto a PIL image.
- **Conversion and Storage**: Rendered images are converted back to NumPy arrays and stored in the images_with_text list for further processing.

Conclusion:

The code successfully demonstrates the implementation of a system for text detection, translation, and overlaying translated text into document images. By leveraging OCR, machine learning, and NLP techniques, the system enables automated extraction, translation, and visualization of text content from diverse sources. Further enhancements and optimizations can be explored to improve the system's accuracy and performance in real-world applications.

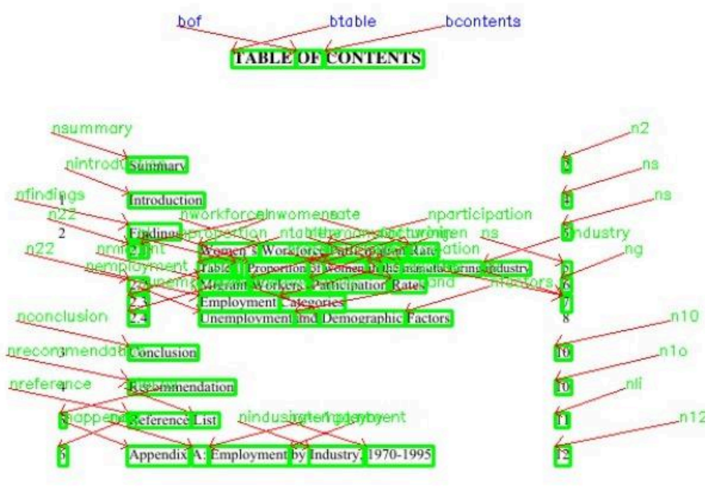


TABLE OF CONTENTS
Summary
Introduction
Findings
Women's Workforce Participation Rate
Table 1 Proportion of Women in the manufacturing industry
Migrant Workers' Participation Rates
Employment Categories
Unemployment and Demographic Factors
Conclusion
Recommendation
Reference List
Appendix A: Employment by Industry, 1970-1995

Translation By Using Google Translate API

HINDI:

मैज़ का अंतर्वस्तु

	सारांश		2
1	परिचय		एस
2	जाँच - परिणाम		एस
	22एल	महलिाएं कर्मचारियों कीभागलेना दर मेज़ एल्सनुपात का औरत में उत्पादन उद्योग	एस
	22	प्रवासी कॉंटिदार भाग लेना दरें	जी
	25	रोज़गार श्रेणियाँ	ठीक
	24	बेरोजगारी और जनसांख्यिकीय कारकों	
3	नष्किर्ष		10
4	सफिारशि		1ओ
एस	संदर्भ सूची		वह
जी	अनुबंध	जैसरोज़गार द्वाराndusiry gto1ggs	12

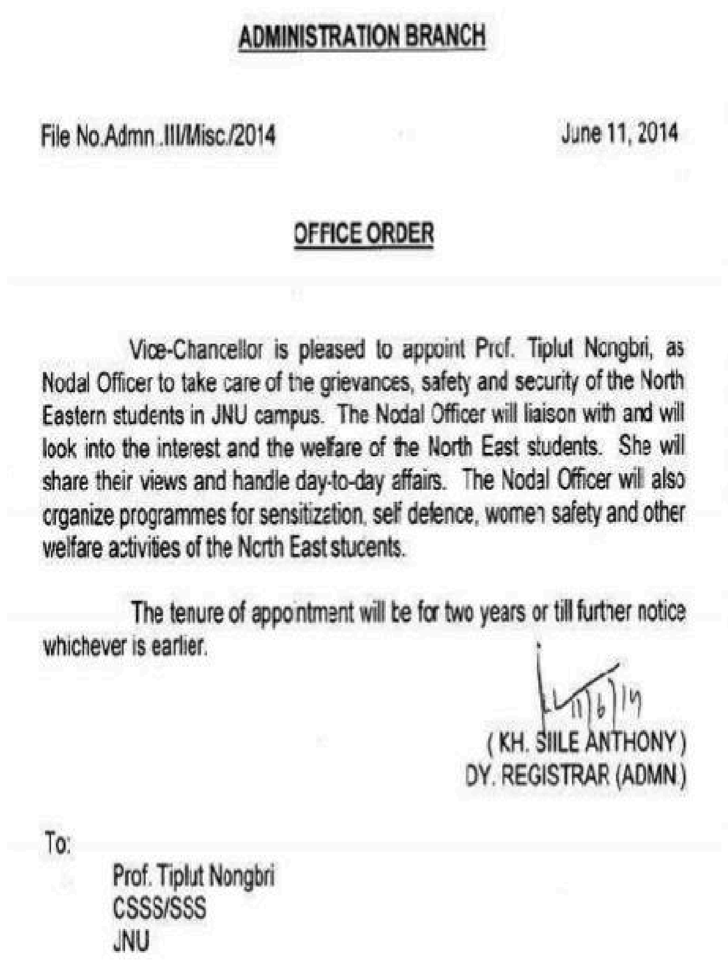
FRENCH:

tableau de Contenu

	résumé		2
1	introduction		s
2	résultats		s
	22l	aux femmesla main d'oeuvreparticipation taux tableau1 proportiondefemmesdans les fabrication industrie	s
	22	migrant fourchettesparticipation les taux	g
	25	emploi catégories	t
	24	chômage et démographiquefacteurs	
3	conclusion		10
4	recommandation		1o
s	référence liste		que
g	annexe	coremploi parindustrie gto1ggs	12

Testing Data

After Translating



ADMINISTRATION BRANCH

File No.Admn.III/Misc./2014 June 11, 2014

OFFICE ORDER

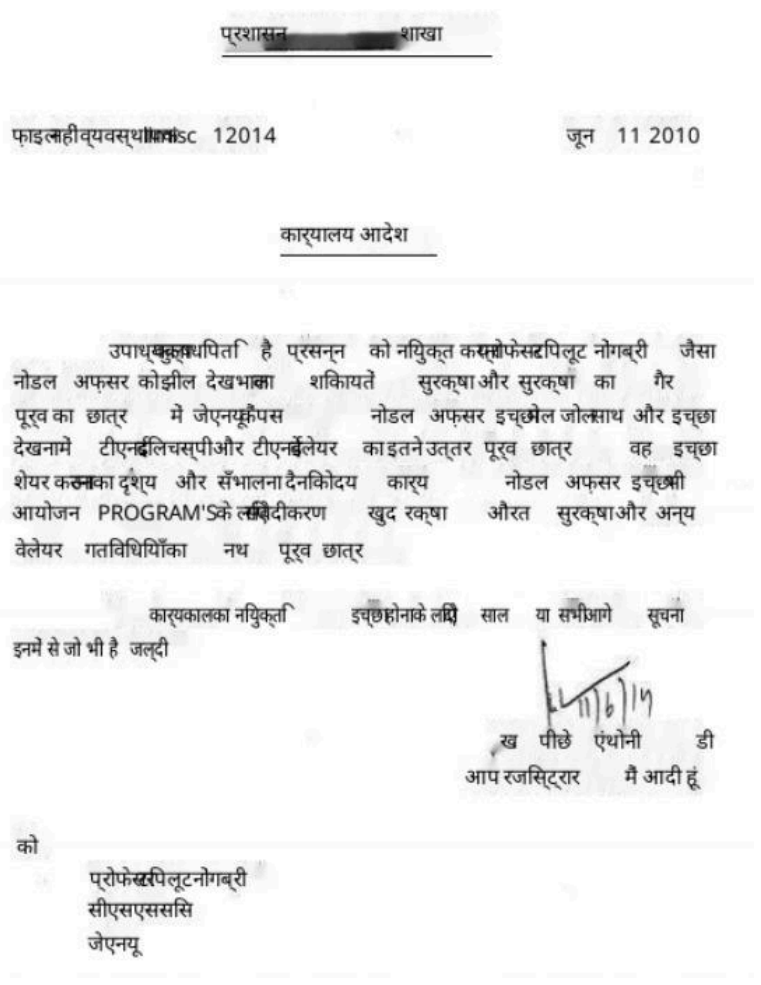
Vice-Chancellor is pleased to appoint Prof. Tiplut Nongbri, as Nodal Officer to take care of the grievances, safety and security of the North Eastern students in JNU campus. The Nodal Officer will liaison with and will look into the interest and the welfare of the North East students. She will share their views and handle day-to-day affairs. The Nodal Officer will also organize programmes for sensitization, self defence, women safety and other welfare activities of the North East students.

The tenure of appointment will be for two years or till further notice whichever is earlier.

[illegible] 11/6/14
(KH. SIILE ANTHONY)
DY. REGISTRAR (ADMN.)

To:

Prof. Tiplut Nongbri
CSSS/SSS
JNU



प्रशासन [redacted] शाखा

फ़ाइलनहीव्यवस्थाisc 12014 जून 11 2010

कार्यालय आदेश

उपाध्यकुलाधपिति है प्रसन्न को नियुक्त करप्रोफेसरटपिलूट नोंगब्री जैसा नोडल अफ़सर कोझील देखभाल शकियतें सुरक्षा और सुरक्षा का गैर पूर्व का छात्र में जेएनयूकैंपस नोडल अफ़सर इच्छमेल जोलसाथ और इच्छा देखनामें टीएनर्दलिचस्पीऔर टीएनर्वेलेयर का इतने उत्तर पूर्व छात्र वह इच्छा शेयर करनाका दृश्य और सँभालना दैनकिोदय कार्य नोडल अफ़सर इच्छभी आयोजन PROGRAM'Sके लवेदीकरण खुद रक्षा औरत सुरक्षा और अन्य वेलेयर गतविधियाँका नथ पूर्व छात्र

कार्यकालका नियुक्ति इच्छहोनाके लदि साल या सभीआगे सूचना इनमें से जो भी है जल्दी

[illegible] 11/6/14
ख पीछे एंथोनी डी
आप रजसि्ट्रार मैं आदी हूं

को

प्रोफेसरपिलूटनोंगब्री
सीएसएससि
जेएनयू