AI based Tender Rate Comparator

Introduction

Tender Committee is a vital procedure in railway tenders and contracts. It involves detailed and exhaustive analysis of tenders and is a time-consuming process. Much of the work of TC is highly objective in nature. For all types of BOQ tenders, it can be divided into three major categories.

1. Document verification(Technical eligibility): Verification and validation of the documents uploaded by both the Tender calling authority (such as estimates, financial and administrative approvals etc) and the tenderers (EMD, technical and financial eligibility etc).

2. Detailed Rate analysis(Financial eligibility): It involves detailed analysis of rate quoted by the selected and approved tenderer so that best rates are brought into working for benefit of railways.

3. Decision: It involves the recommendation of TC and accepting of TC recommendations by Tender accepting authority for successful awarding of contracts in the benefit of railways.

Concentrating of S.no-02: 'Detailed Rate analysis', it is to be noted that it is a time consuming process. My method, in this document proposes an AI based toolkit to help the Officials in this part so as to improve TC performance. Any time savings in this part will be very helpful for the TC and improve its performance.

IREPS:

This proposed methodology requires users to be able to login to IREPS for downloading the documents correctly. The following are the system requirements for using this tool:

1. Windows 10 or Windows 11 PC with atleast 10GB free space in C drive.

2. 16 GB RAM.

3. Microsoft Excel 2013 or greater.

4. Visual studio latest version with C++ development module.

5. Ghostscript(depending on user system, win 32 bit or 64 bit). URL: https://github.com/ArtifexSoftware/ghostpdl-downloads/releases

6. Good internet connection for initial installation.

7. Good GPU of atleast 8GB for quick running of AI models.

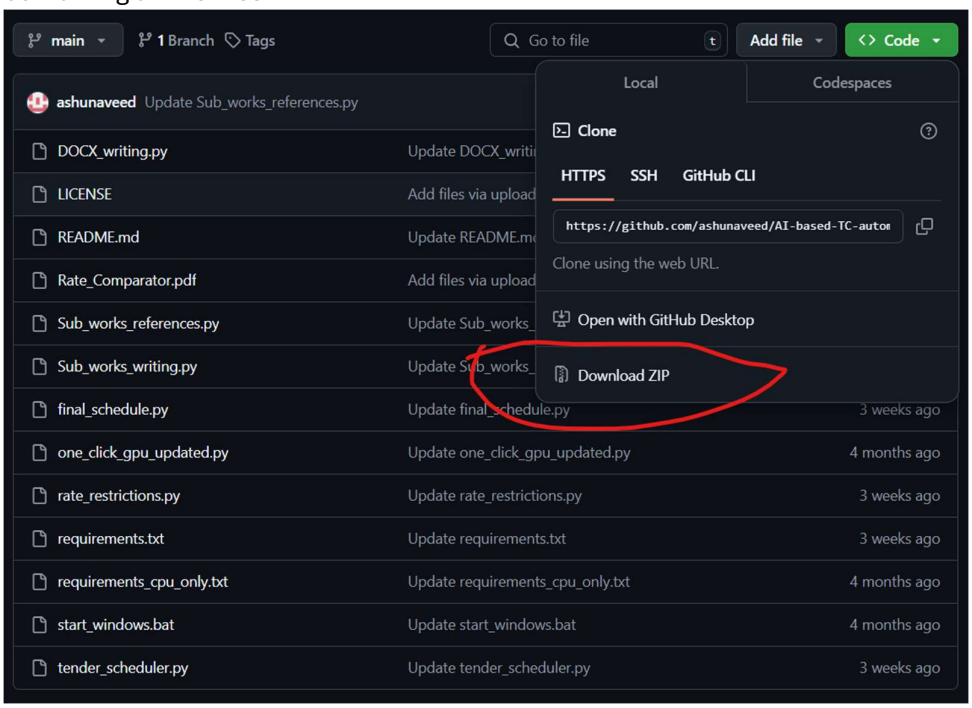
8. The AI model can be downloaded from: https://huggingface.co/lmstudio-community/Meta-Llama-3.1-8B-Instruct-GGUF/resolve/main/Meta-Llama-3.1-8B-Instruct-Q8_0.gguf?download=true

Input documents requirements: The financial bid files and necessary reference LAR files, PO references etc should be downloaded from IREPS in HTML format (please open the

financial bid/LOA in interactive(HTML) format and Purchase orders in PDF format to be saved in a convenient folder .

Detailed process flow:

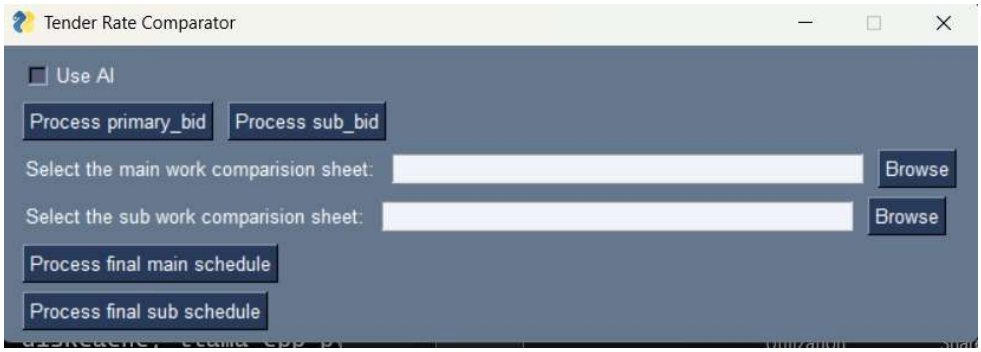
1. Download the files from Git-hub, extract the files and save them to a folder on your computer, say with name 'AI-based-TC-Automation'(The name of the folder should not contain any spaces). The below image shows the image of github page containing all the files.



2. Double click the start_windows.bat file. If windows prompts about unverified source, click 'run anyway'. During first installation, the program will create a python environment and download the required files (requires internet) and save them. The initial installation is quite large(~10GB) and the user should have a reliable internet connection for downloading ~10GB data.

3. After successful installation of the files, a GUI as shown will pop up on the screen. This is the main GUI and indicates successful installation.

4. Detailed description of the actions is as follows:



The user will have to select the checkbox: 'use AI' to use the AI model in the processing of primary bid and sub bids. The AI model 'Llama-3.1-8B' used in this work is a advanced 8Billion parameters. AI models of size 70B and above substantially improve the performance and accuracy of the matching but they require quite high processing, RAM and GPU processing.

1. 'Process primary_bid': This button is to be clicked when the user wants to evaluate the primary/ main bid of the tenderer. After clicking this, a new popup is generated which asks whether this is an initial bid or after a negotiation, clicking of which the user will again be prompted to click whether this work is completely related to Civil Engineering department. The user has to click buttons correctly. Then a new popup is generated that asks user to upload the required bid files and submit it. Then user has to upload the necessary LAR files (in .html format) and PO files (in .pdf format)(For the PO's, it will select basic rate and add taxes if any. It will not select transportation charges). Then the modules will evaluate the bid (This module will evaluate the rates item wise only.) and create and save a excel sheet named main work comparison sheet followed by date and time of generation in the main_work_comparision folder. If there are any rate restrictions written in the LARs, it would be displayed in the first cell of each LAR. The user has to edit the excel sheet for any corrections such as wrong matches, if any rate restrictions on future usages etc and save it. On a good commercial computer without a GPU, for comparing a bid of 200 non SOR items with 10 LARs each of 200 items, it would take around approximately 6-8 hours.

2. 'Process sub_bid': This button is to be clicked when user has to process any sub bids/schedules. After clicking this button, the user has to input the number of sub bids that are being dealt with and upload the bids and reference LARs and POs in correct format. This module will evaluate the bid item wise only. For an Civil Engineering work, it will also check whether the LARs are from same schedule of rates also. This module will not work for comparing different schedules such as DSR-2019 and DSR-2020. After evaluation, a new excel sheet is created and saved in the sub_work_comparision folder which will contain the bids and LAR comparison with one bid and its comparison LARs per sheet. Rate restrictions if any would be written in the last cell for each LAR in each sheet. User will have to make required corrections if any such as wrong matches, if any rate restrictions on future usages etc. and save it. On a good commercial computer without using GPU, for comparing a bid of 200 non SOR items with 10 LARs each of 200 items, it would take around approximately 6-8 hours.

3. 'Select the main work comparison sheet': After clicking the 'Browse' button beneath this label, the user will be prompted to upload the main work comparison sheet generated in step 1 and corrected by user. Then user will have to click 'Process for final

evaluation of main schedule' button, clicking of which the user will be prompted to enter how much variation is he willing to accept from the advertised rates(slab1) and average of LAR rates(slab2) respectively. After submission of these percentages, the module will process the main work comparison sheet such that if the quoted rate in percentage from advertised rate and average of LAR rates is less than both the slabs then it will recommend for acceptance of rates and if it is greater than any of the slabs, it will recommend for one round of negotiation. A new excel sheet is generated named 'final_main_schedule' followed by date-time of creation of the file in the Main_work_schedule_report folder. The user can make necessary corrections and upload it on the IREPS.

4. 'Select the sub work comparison sheet': After clicking the 'Browse' button beneath this label, the user will be prompted to upload the excel sheet created in step 2, 'Process sub bid' after necessary corrections. After uploading, the user will have to click 'Process for final evaluation of sub_work_schedule_report' button, clicking of which the user will be prompted to upload quantities such as how much net variation(including the escalation given in tender schedule on the whole and rebate if any) the tenderer has quoted and how much escalations from advertised rates and LAR rates can be accepted. Then a new excel sheet is generated named 'final_sub_schedule_report' followed by date-time of creation of the file in the sub_work_schedule_report folder. The user can make necessary corrections and upload it on the IREPS.

Background Processes:

1. This module utilizes Miniconda, Python and modules such as Pandas, PySimpleGUI, AI and NLP libraries and Camelot-py for evaluation. In the background, for each reference LAR uploaded by the user, writes the LAR name in the first row and for each item in the bid compares each item in the LARs using latest version of text distance module and picks up the items of maximum match (only if that maximum is above 60%) and after interacting with an AI model for complete and exhaustive enquiry about the items in question, it saves the item name, its rate, its S.no in the appropriate cells. If the user finds that an item is wrongly matched or its rate is restricted for future references, it can be deleted appropriately.

2. If the user feels manual updating is required or some item rate should be added from a LAR (very old LOA or PO or BQ), he can do so by creating a column named after LAR name and writing the S.no and rate in following format and save it. Eg. If an item from LAR named 'BQ from RRR enterprises' is to be added in comparison sheets, the name of the LAR should be written in a cell in first row beside the previous LAR names and the S.no and rate should be written at S.no row of the correct item in 'S.no in BQ $#$ rate' format. The rate should be only in numerical format i.e. 'S.no-21 $#$ 25.45'. Please note that the string '$#$' is the identifier used for bifurcation of the string into serial number and rate. If only rate is to be written, it can be written in numerical format at appropriate S.no. and saved. The final bid comparator will accept the changes.

3. For item comparison more efficient and robust hugging face models and transformer models can be used but it will be more time and power consuming with good increments in accuracy of matching of items.

Future updates:

The module will be updated for including more features such as

1. document verifications,
2. decision making ability,
3. More efficient AI, NLP algorithms execution in the future.

So I request the everyone to regularly visit the Git-hub page for any updates.

Acknowledgements: I would like to express my deepest gratitude to PCSTE/SWR, CAO/CN/BNC, CSTE/CN/BNC, DRM/MYS, ADRM/MYS, Sr.DSTE/MYS, Dy.CSTE/CN/BNC and all other senior officers and S&T staff for giving me this opportunity for developing this module.

About the Author: This module is created and developed by Md. Naveed Ashfaq currently working as DSTE/CN/BNC. For any suggestions, please write to naveedashfaq111@gmail.com.
The Git Hub page link is: https://github.com/ashunaveed/AI-based-TC-automation