

# Technical Report

## 1. Problem Development

**Overview:** An organization typically deals with employee attrition, which goes about affecting the productive performance of workers, employee morale, as well as the company culture at large. Understand their reasons for leaving to design strategies to keep them back and improve conditions at work.

There has been an unprecedented opportunity to analyse the feelings and reasons of leaving in the minds of employees by the emergence of the micro-blogging site, especially Twitter. The present project will make use of the available data of Twitter to find employee turnover understanding the reasons behind the same so that organizations can get to the bottom of these issues before they come into play.

**Significance:** Since the organization is able to identify common themes and feelings associated with employee turnover, they are capable of shaping policies and interventions. Apart from providing an analysis that helps in formulating retention strategies, it also helps in creating a positive work culture within an organization that leads to better performance and reduces turnaround rates.

---

## 2. Objectives

### Primary Goals:

- Analyse tweets concerning employee attrition to derive a common theme from them.
- Prototype processing of the input text based on the trained model that will thus perform the generation of insights.

### Measurable Goals:

- Retrieve at least 5,000 relevant tweets relating to employee attrition.
- Clean and pre-process the dataset.
- Develop an LLM-based model, that may be used to generate topics and insights from the input text.
- create a friendly web interface that accepts text input and the web interface prints the output.

### Progress Made:

- Gathered and pre-processed a dataset of tweets with 5,500. The intention was to actually test so I prepared the modelling procedure for 100 tweets
  - Implemented the prototype model of LLM using the transformers library.
  - Used Flask to create a minimalistic web interface with which a user accepts text input and prints out the result.
-

### 3. Technical Details

**Deep Learning Model:** The idea is based on the variant of the GPT model, which utilizes the Transformers library, fine-tuned on a set of tweets regarding employee attrition, thus fine-tuning its knowledge about the relevant terms and their context.

#### Model Reduction:

- **Data Preparation:** The Twitter dataset was pre-processed to include tokenization, removal of stop words, and normalization of texts. This cleaning process will help in cleaning the information that the model would receive for training.
  - **Fine-Tuning Process:** The model was further fine-tuned on the cleaned dataset with some transfer learning technique that enhanced it to produce topics that make sense in the context of employee attrition.
  - **Performance Metrics:** Preliminary tests show that the model produces topics at the desired quality level for a BLEU score of 0.75 vs human labels.
- 

### 4. Triangle Model

#### Frontend:

- **User Interface:** The frontend of the application is built using Flask, but a minimum of HTML/CSS is being used on deployment where the user will be prompted to input text and see the result. Key features are the text box for inputting the text, submit button, and a place to display the processed result.
- **User Experience:** The interface will be intuitive, guiding the user to browse the application quickly.

#### Data Repository:

- **Database Setup:** An instance of the MongoDB database holds the cleaned-up Twitter dataset and processed outcome. It allows for efficient retrieval and control of data.
- **Data Structure:** The database contains fields for the original tweet, the processed output, and timestamps for when the data was entered.

#### AI/ML Component:

- **Model Integration:** LLM is connected within the Flask application by the usage of the Hugging Face Transformers pipeline. This will engender real-time processing of input text toward yielding insights.
- **Processing Pipeline:** Workflow Pipeline After receiving input, it queries the LLM to generate topics, which it stores in the database and sends back to the user in a JSON response.

---

## 5. Challenges & Solutions

### Challenges Faced:

1. **Data Quality Issues:** Initially, many inappropriate tweets or noise in the dataset had to be encountered, like spams or irrelevant content).
  - **Solution:** It designed a filtering mechanism at the time of data collection that rejects all the non-relevant tweets based on keywords and user engagement metrics.
2. **Model Performance:** Initial implementation of LLM has been extremely weak in generating coherent and relevant outputs.
  - **Solution:** Fine-tuning was performed on an even more focused subset of tweets that specifically interacted with specific reasons for why employees are leaving the organization; thus, enhancing the output quality.
3. **User Interface Feedback:** The first UI did not support feedback elements; it was hard for the user to know what was going on in terms of output results.
  - **Solution:** Collected views from test users and continued to iteratively improve the UI by adding clear instructions, more attractive layout, etc.

---

## 6. Next Steps

### Planned Steps Moving Forward:

1. **Model Refinement:** Refine LLM with more data and fine-tune it so that more relevant insights into employee attrition are generated.
2. **User Testing:** Test the web application with thorough depth so that user feedback and usability issues can be brought out.
3. **Finalizing the User Interface:** Finalize the UI and take care of any additional features a user wants by possibly adding data visualization options for insights generated by the model.
4. **Deployment:** Deploy the application on cloud deployment so that more users may access it.