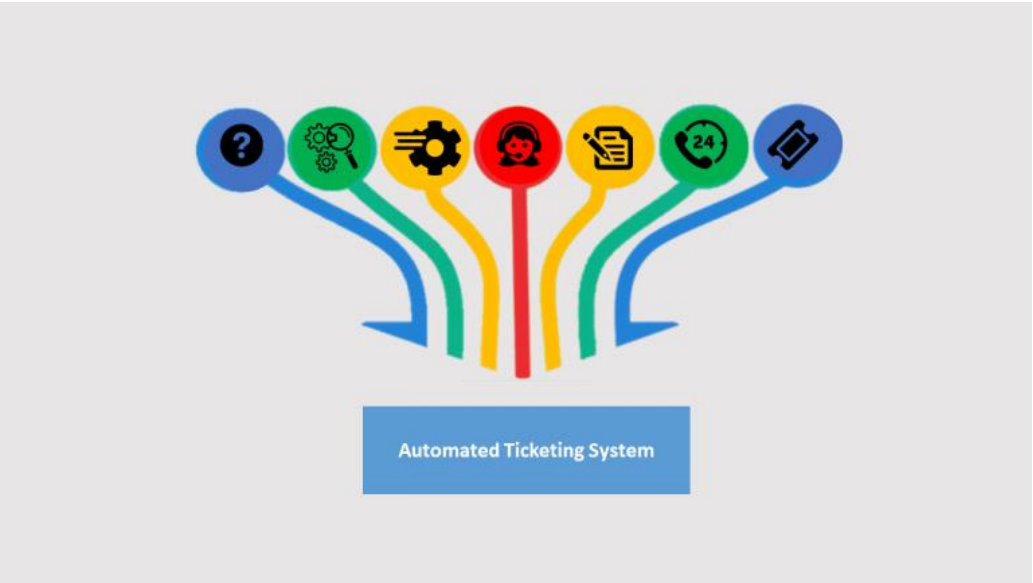


INTERIM REPORT

Automatic Ticket Assignment

POST GRADUATE PROGRAM IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Capstone Project

Atul

Akshay Mathur

Aaby Sivakumar

Siva Kumara N V

Teja R K L

The institution

GREAT LAKES

Great Lakes e-Learning

greatlearning
Learning for Life

Colaboration with

TEXAS McCombs
The University of Texas at Austin
McCombs School of Business

Table of Contents

1. Understanding the business

In any IT industry, Incident Management plays an important role in delivering quality support to customers. An incident ticket is created by various groups of people within the organization to resolve an issue as quickly as possible based on its severity. Whenever an incident is created, it reaches the Service desk team and then it gets assigned to the respective teams to work on the incident. The Service Desk team (L1/L2) will perform basic analysis on the user's requirement, identify the issue based on given descriptions and assign it to the respective teams.

The manual assignment of these incidents might have below disadvantages:

1. More resource usage and expenses.
2. Human errors - Around ~25% of Incidents are wrongly assigned to functional teams.
3. Delay in assigning the tickets
4. More resolution times
5. If a particular ticket takes more time in analysis, other productive tasks get affected for the Service Desk

If this ticket assignment is automated, it can be more cost-effective, less resolution time and the Service Desk team can focus on other productive tasks.

2. Problem Statement

One of the key activities of any IT function is to “Keep the lights on” to ensure there is no impact to the Business operations. IT leverages Incident Management process to achieve the above Objective. An incident is something that is unplanned interruption to an IT service or reduction in the quality of an IT service that affects the Users and the Business.

The main goal of Incident Management process is to provide a quick fix / workarounds or solutions that resolves the interruption and restores the service to its full capacity to ensure no business impact. In most of the organizations, incidents are created by various Business and IT Users, End Users/ Vendors if they have access to ticketing systems, and from the integrated monitoring systems and tools.

Assigning the incidents to the appropriate person or unit in the support team has critical importance to provide improved user satisfaction while ensuring better allocation of support resources. The assignment of incidents to appropriate IT groups is still a manual process in many of the IT organizations. Manual assignment of incidents is time consuming and requires human efforts.

There may be mistakes due to human errors and resource consumption is carried out ineffectively because of the misaddressing. On the other hand, manual assignment increases the response and resolution times which result in user satisfaction deterioration / poor customer service.

3. Objective

From the given problem description, we could see that the existing system is able to assign 75% of the tickets correctly. So, our objective here is to build an AI-based classifier model to assign the tickets to right functional groups by analyzing the given description with an more accuracy.

4. Observations from the given Dataset

- Four columns – Short Description, Description, Caller and Assignment group
- 74 Assignment groups found - Target classes
- Caller names in a random fashion (may not be useful for training data)
- European non-English language also found in the data
- Email/chat format in description
- Symbols & other characters in the description
- Hyperlinks, URLS & few image data found in the description
- Blanks found either in the short description or description field
- Few descriptions same as the short description
- Few words were combined together
- Spelling mistakes and typo errors are found

Observations from Target Class

- The Target class distribution is extremely skewed.
- A large no of entries for GRP_0 (mounting to 3976) which account for ~50% of the data.

5. Data Pre-processing

Below steps have been performed for initial pre-processing and cleanup of data.

- Replaced Null values in Short description & description.
- Dropped the caller field as the data was not found to be useful for analysis
- Contraction words found in the merged Description are removed for ease of word modelling
- Changed the case sensitivity of words to the common one
- Removed Hashtags and kept the words, Hyperlinks, URLs, HTML tags & non-ASCII symbols from merged fields.
- Translating all languages (German) to English
- Tokenization of merged data
- Removal of Stop words
- Lemmatization
- WordCloud created for all available 50 groups to have more information specific to Assignment groups
- Created Plot to understand the distribution of words

The target class is extremely skewed data. The target class were filtered for less than 10 entries and grouped together as misc_grp as there is no much information with the groups individually.

6. Data Cleansing activities for Data Preparation

Once the data pre-processing is done, the data cleansing is carried out to make it ready for processing. The various data cleansing activities are mentioned in detail in the section below:

- A function has been created to clean up the unwanted information from initial observations.
- All the contractions have been removed
- All words have been converted to lowercase. The email headers and sender information are removed

- All the numbers, non-dictionary characters, newline characters, hashtag, HTML entities, hyperlinks, extra spaces and unreadable characters have been added to the function. Ensured to remove any caller names included in the description column.
- The function is applied to the Description column and data is generated for further analysis.
- Lemmatization & Stop words removal
    - Stop words have been removed using nltk corpus modules.
    - Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item.
    - Lemmatization is like Stemming but it brings context to the words. So, it links words with similar meanings to one word.
    - Here we have preferred Lemmatization over Stemming because lemmatization does morphological analysis of the words.

Lemmatized descriptions based on assignment groups is shown below.

Assignment group	lemmatized
GRP_0	user details employee manager name user name a...
GRP_1	event critical value mountpoint threshold toda...
GRP_10	receive fail receive fail receive fail receive...
GRP_11	hello service nee monitor manufacture drawing ...
GRP_12	c label server space consume space available g...
GRP_13	receive fail receive two customer account get ...
GRP_14	intermittent service configair server require ...
GRP_15	hi channel partner email address djhadkudhd re...
GRP_16	receive cid bwfhtumx japznrvb regional control...
GRP_17	employee get error user authentication fail tr...
GRP_18	receive hello team could please generate deliv...
GRP_19	unable take print xdvwitpm zscxqdhoalaramdntya...
GRP_2	try change password acc attach work office vac...
GRP_20	datum correctly pull employee attendee interfa...
GRP_21	need approve new product request internal user...
GRP_22	assign crm license nyrjkctu use profile like q...
GRP_23	unable see current course ethic user -PRON- wo...
GRP_24	support fa r ohxdwngl hallo es ist erneut pass...
GRP_25	crashes confirmation delete able remove scrap ...

Analysis using WordCloud

WordCloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word Clouds have been generated with All available words & top 100 words.

We have also inferred few observations over the target class – Assignment groups with word clouds for top 50 words from each group. Word clouds for all words and top 100 words are as shown below.

7. NLP Data Augmentation

Data Augmentation is used to boost the model performance by various techniques. The following techniques were used for the augmentation of the data as part of this project.

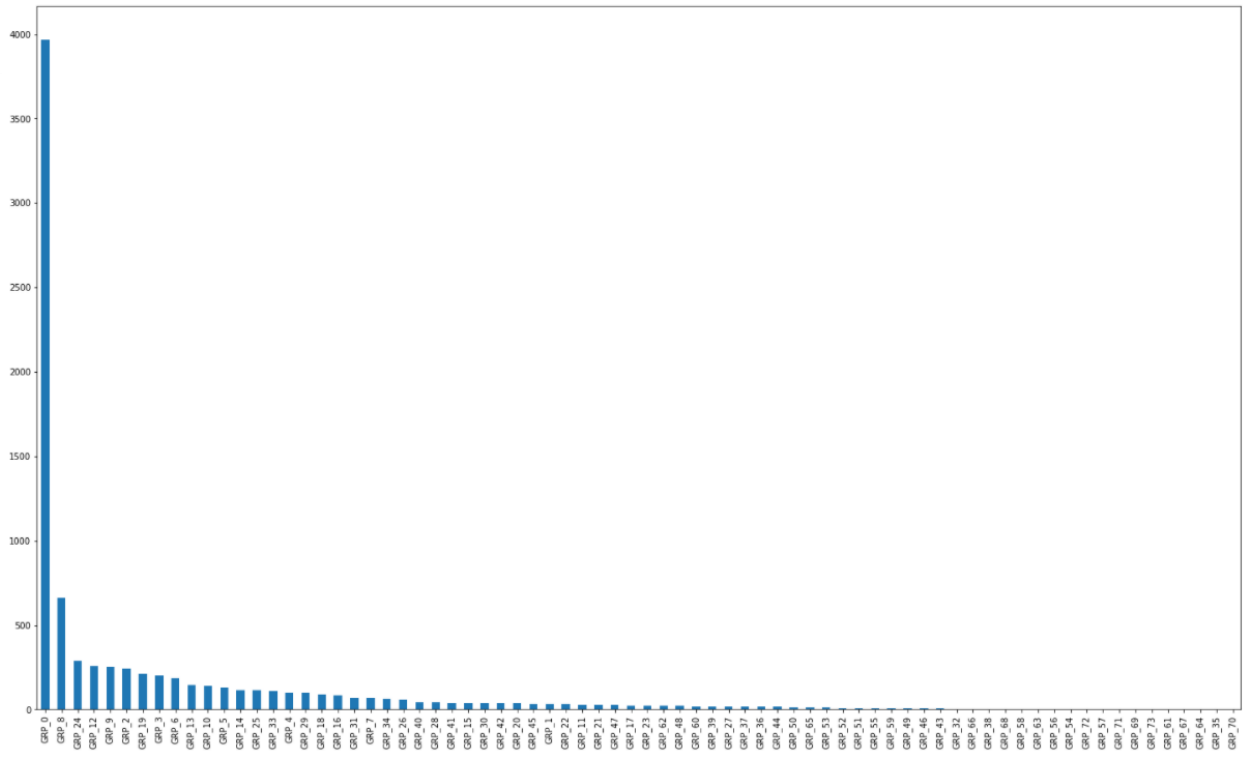
As a first step, the stop words are listed which should be avoided from further data processing as they are the most common words and do to have much impact on the overall result. One more level of cleansing is done to remove the hyphens and extra spaces.

Then the following augmentation techniques are used.

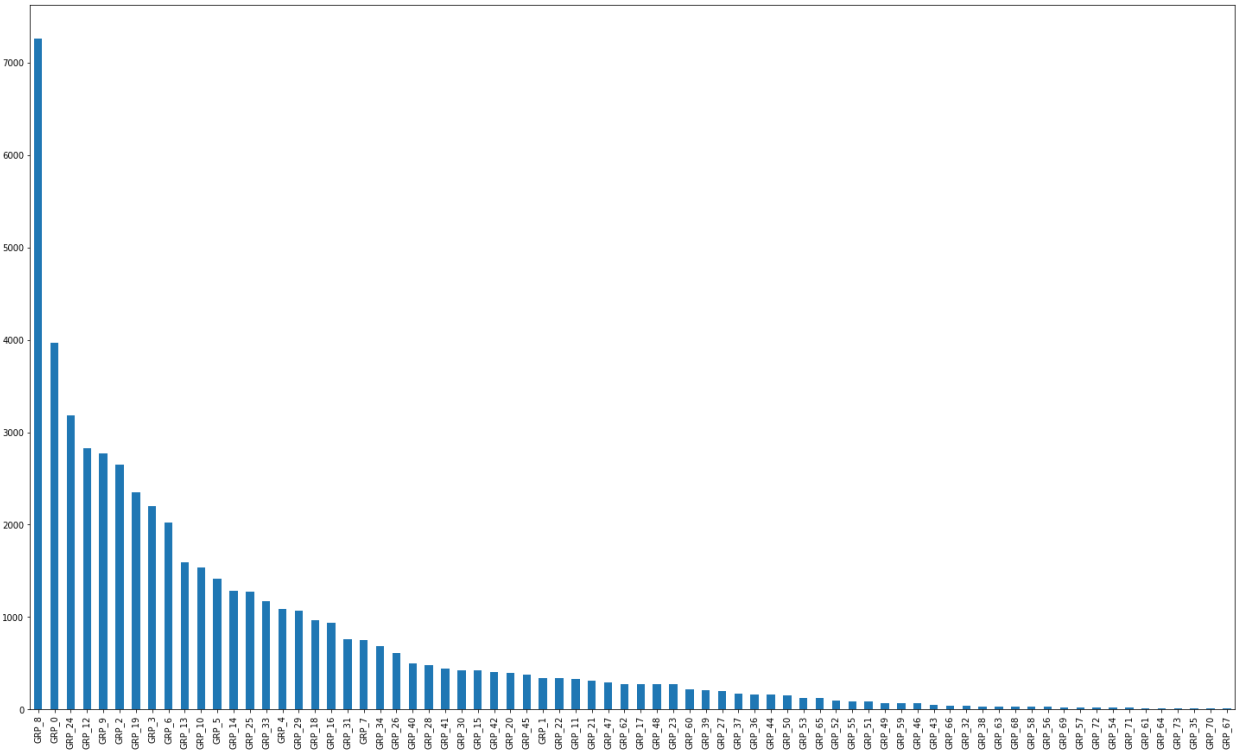
a. The random words are replaced by its synonyms. The synonyms are collected from wordnet.
b. Random deletion to delete some random words from the sentences
c. Random swap by swapping two words randomly in a sentence
d. Random insertion of words into the sentence

A main augmentation function is created, and the different modules for augmentation is called from the function to execute the required technique.

The classification plot before the augmentation is as below



The classification plot after augmentation is given below.



8. Feature Engineering – Embedding Techniques - BERT

BERT stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks.

- BERT is based on the Transformer architecture.
- BERT is pre-trained on a large corpus of unlabelled text including the entire Wikipedia(that's 2,500 million words!) and Book Corpus (800 million words).
- BERT is a "deeply bidirectional" model. Bidirectional means that BERT learns information from both the left and the right side of a token's context during the training phase.

Maximum length of the sequence is taken as 50.

Feature Matrix is created with ids, tokens and sequences for training and test data.

We are using the pre-trained model "bert-base-uncased". The model creates based on BERT embedding is as follows:

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_idx (InputLayer)          [(None, 50)]         0
__________________________________________________________________________________________________
input_masks (InputLayer)        [(None, 50)]         0
__________________________________________________________________________________________________
input_segments (InputLayer)     [(None, 50)]         0
__________________________________________________________________________________________________
tf_bert_model_1 (TFBertModel)   ((None, 50, 768), (N 109482240   input_idx[0][0]
                                                                 input_masks[0][0]
                                                                 input_segments[0][0]
__________________________________________________________________________________________________
global_average_pooling1d (Globa (None, 768)          0           tf_bert_model_1[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 64)           49216       global_average_pooling1d[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 74)           4810        dense[0][0]
==================================================================================================
Total params: 109,536,266
Trainable params: 54,026
Non-trainable params: 109,482,240
```

9. Next steps

- Fit the model with training data and check performance of the model for both training and test data.
- Tuning hyperparameters for optimized performance of the model
- Check above again post merging of similar groups and report if any gains in performance is achieved
- Explore other embedding options including the possibility of leveraging custom embedding
- Explore other classification models and tune them and have a comprehensive comparative analysis report
- Like custom embedding, in our case we can check the possibility of having our model written completely by ourselves instead of pre-trained model.