Capstone Project

Interim project report

Automatic Ticket Assignment using NLP

Submitted by:
NLP Group 7

Contents

1. Introduction

In IT support industry, the key aspect to keep the business inline & to grow further is the support/service provided to resolve technical issues. In that Incident management is a key point that ensures the operation in line. In large organizations, assignment of support/service ticket to appropriate group is still manually performed.

The main goal of Incident Management process is to provide a root cause/analysis, quick fix / workarounds or solutions that resolves the interruption and ensures completion of work in planned time but manual classification of IT service desk tickets may result in routing of the tickets to the wrong resolution group, it leads to reassignment of tickets, unnecessary resource utilization and delays in resolution time.

To overcome above issues, machine learning algorithms can be used to automatically classify the IT service desk tickets. Service desk ticket classifier models can be trained by mining the historical unstructured ticket description and the corresponding label/group. After processing the unstructured data using empirically developed methodology that is data pre-processing, words stemming, feature engineering, training classifier model and machine learning algorithm tuning. The model can then be used to classify the new service desk ticket based on the ticket description. It utilizes an accurate ticket classification machine learning model to associate a support/service desk ticket with its correct service from the start and hence minimize ticket resolution time, save human resources, improved productivity, customer satisfaction and growth in business.

2. Summary of problem statement, data and findings

This section details about the **“Automatic Ticket Assignment”** NLP problem statement. We will discuss the dataset used for this project and will present a few findings from this data.

2.1. Problem statement summarization

Incident management is critical to every software organization. Users typically face issues while using the software and resolution time of such issues is a critical parameter in assessing the software usability.

Typically the incidents are reported to the service desk, which analyzes the problem and assigns to concerned L1/L2 support groups. This activity is manual and error prone and hence it takes a long time for resolution of user reported incidents.

To reduce this cycle time it is proposed to employ the NLP techniques to automatically assign the incoming incidents to the concerned groups based on the description of the incidents. This makes the problem as a classic example of classification based NLP machine learning problem.

2.2. Data analysis and findings

The dataset used for this project can be found at dataset link. Here is some initial analysis -

- The dataset consists of 8500 rows and 4 columns. Each row represents information about an incident. The columns are **Short description**, **Description**, **Caller** and **Assignment group**.
- The Short description column summarizes the incident description and is short text.
- The Description column is the in detail description of the incident detailing the issue.
- The Caller column identifies the user with the firstname and lastname of the user.
- The Assignment group is the support group to which a particular incident was assigned to.

Findings from the data

Target column – The Assignment group is the target column of this problem.

The caller column does not affect the target column directly and hence can be safely removed from the model building process.

3. Summary of the approach to EDA and pre-processing

This section details the data preprocessing and EDA on the dataset.

3.1. EDA

Below are the observations when exploratory data analysis was performed on the given dataset –

- There are null values present for Short description and Description columns in the dataset. Specifically 8 incidents do not have values for Short description and 1 entry does not have a value for Description column.
- There are in total 74 unique support groups named from GRP_0 to GRP_73 present in the database. Analysis shows that there are 3976 (46%) incidents assigned to GRP_0 and the rest 4524 among the rest 73 groups. This makes the data highly disproportionate. The ticket distribution for top 10 assignment groups is shown below -

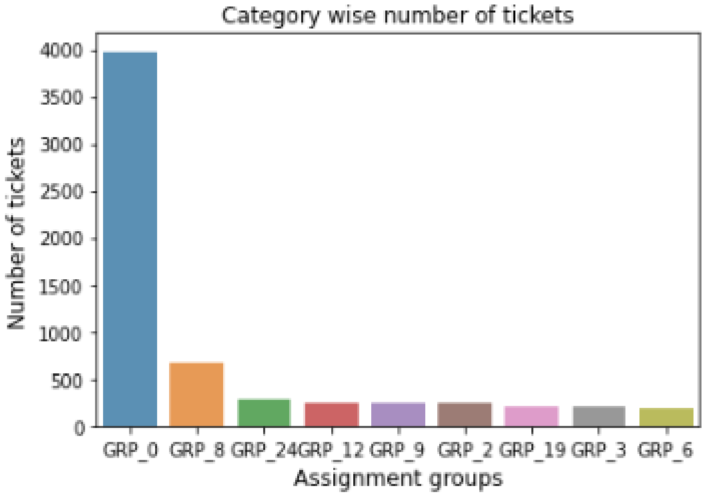


Figure 1 Category wise number of tickets

- There are 2950 unique users that have reported the incidents and the topmost user has 810 (27%) number of incidents raised. The ticket distribution for top 10 users is as shown below -

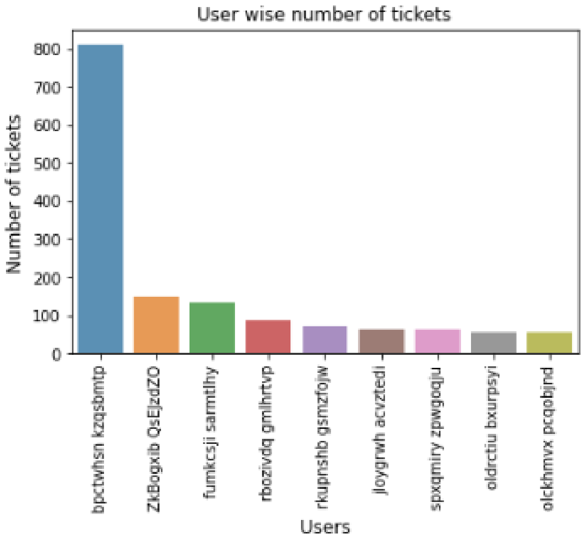


Figure 2 User wise number of tickets

- See Figure 3 for the word cloud data for Short description column –

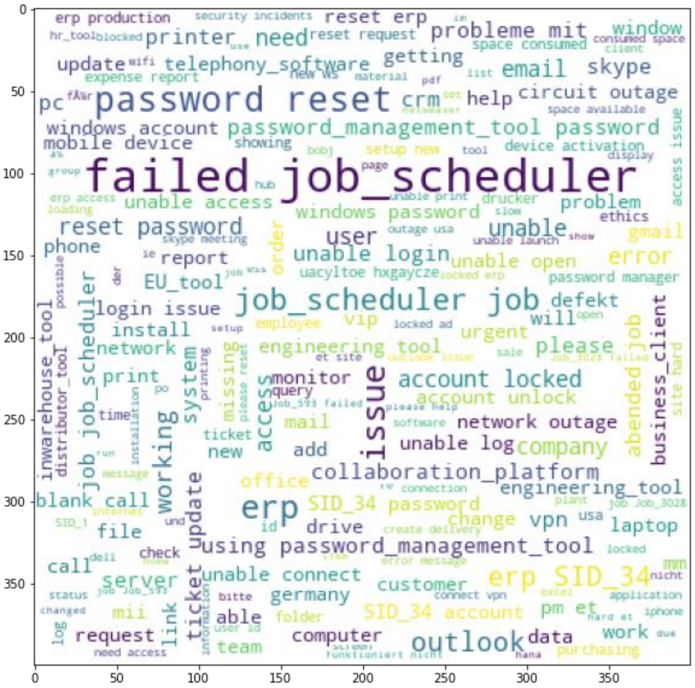


Figure 3 Word cloud for Short description column before preprocessing

- See figure 4 for the word cloud data for Description column –

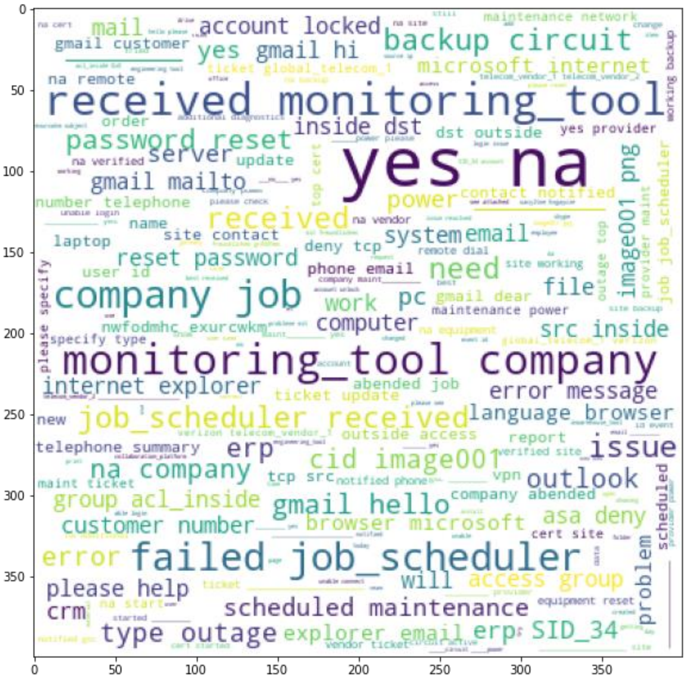


Figure 4 Word cloud for Description column before pre-processing

- The description column for most of the tickets also consists of text like “Reported by: emailid”. This can be removed as callers email id has no relationship with the target column.

3.2. Data preprocessing

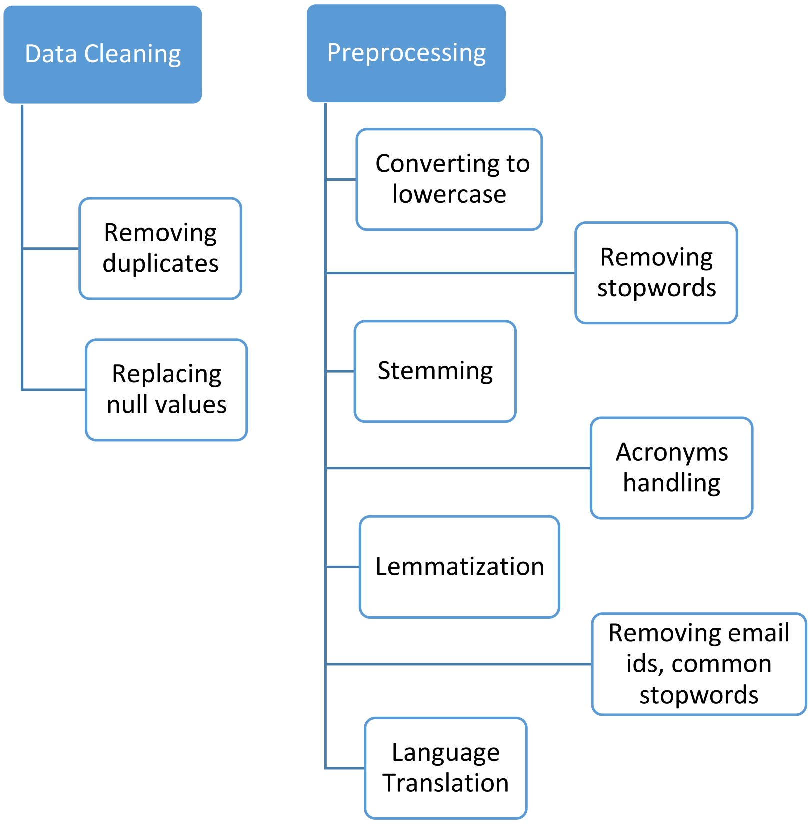


Figure 5 The data cleansing process

The fundamental aspect of machine learning that determines how good the ML model can perform is **Data**. If data is insufficient or is not processed correctly to target the problem, we cannot expect the model to perform well.

We have identified following preprocessing steps for the given dataset –

- The first step is to get rid of null values from the dataset. As noted earlier there are 9 rows that null values for either of Short Description and Description columns. We will replace the null values with stop words.
- There are 83 duplicate entries in the dataset, which can be safely removed.
- As noted earlier, we can get rid of text like “Reported by emailid” from the Description column. To build the email id for each user we can use the first name and last name information from caller column.
- The caller column does not affect the target column and hence can be safely removed in the end.
- The data is highly imbalanced as a lot of entries correspond to GRP_0. Hence we will down sample the dataset for GRP_0. Also there are 6 groups for which only one entry is present.

This needs to be handled appropriately as well.

- The text in the Description column can be converted to all lowercase.
- Stop words can cause a lot of noise during the word embedding’s hence we can remove them as well.
- After data cleansing below is the word cloud data for short description column –

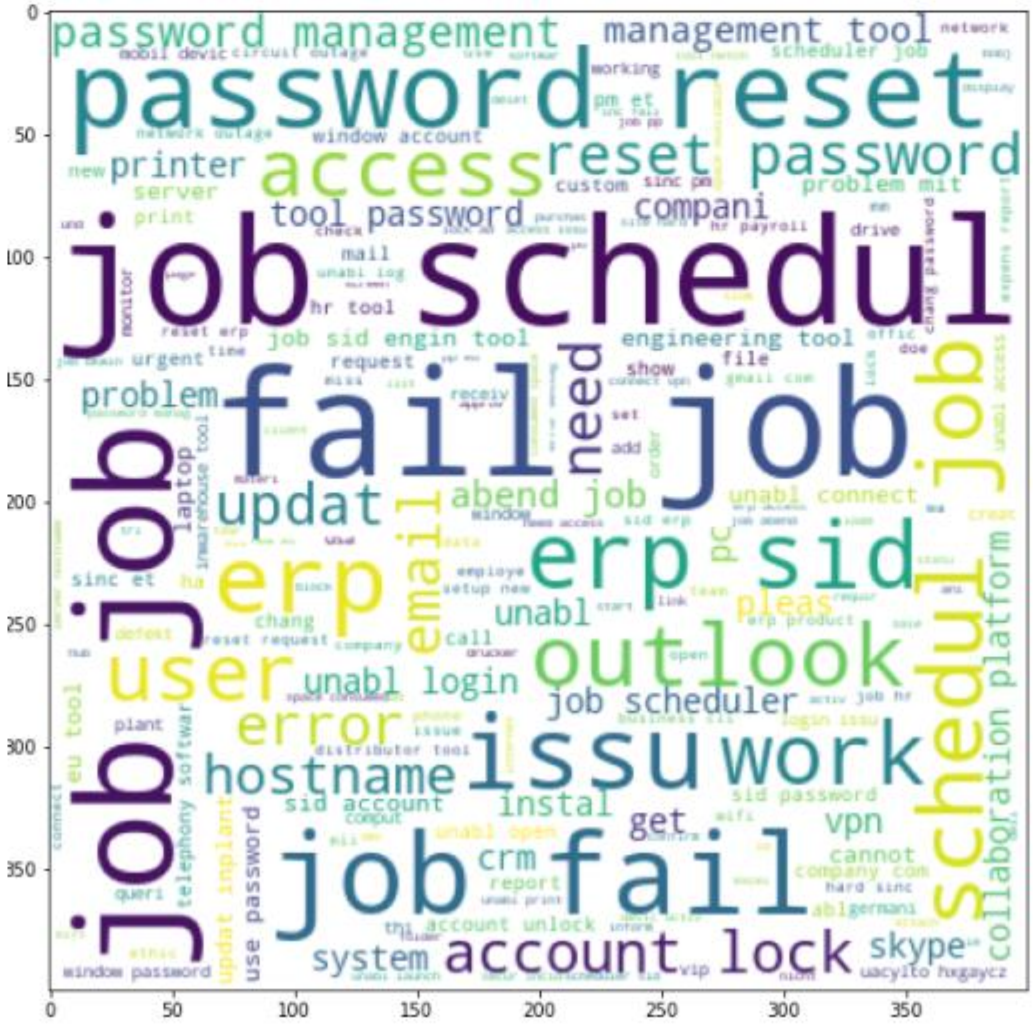


Figure 6 Word cloud for Short description column after pre-processing

- See Figure7 for the word cloud for Description column –

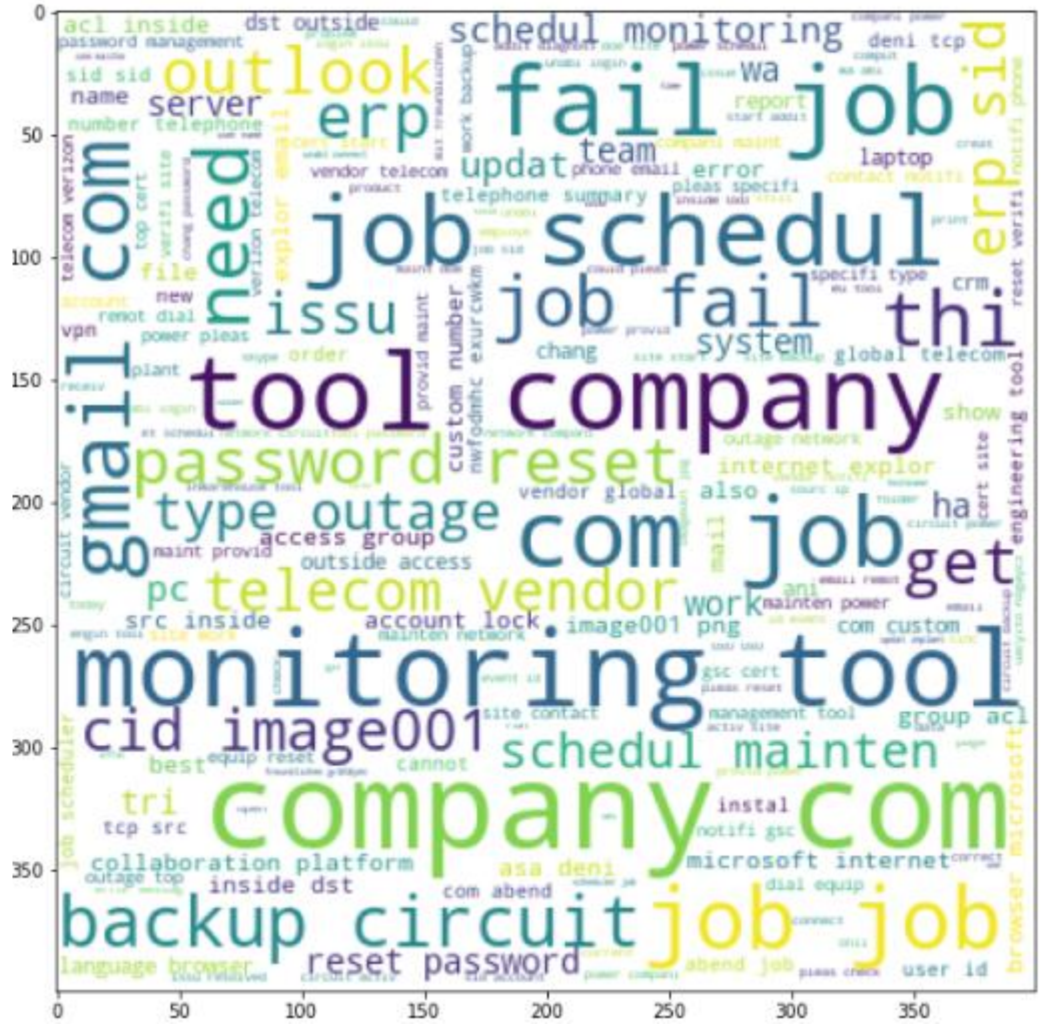


Figure 7 Word cloud for Description column after pre-processing

Dealing with languages other than English

Upon analysis of data, other languages apart from English were found. We used Fasttext library and used a pre-trained model to predict the language from the description of an incident. We have used a threshold of 50% to indicate that we only consider the languages as other than English if the confidence output from model is over 50% value.   Below is the result -

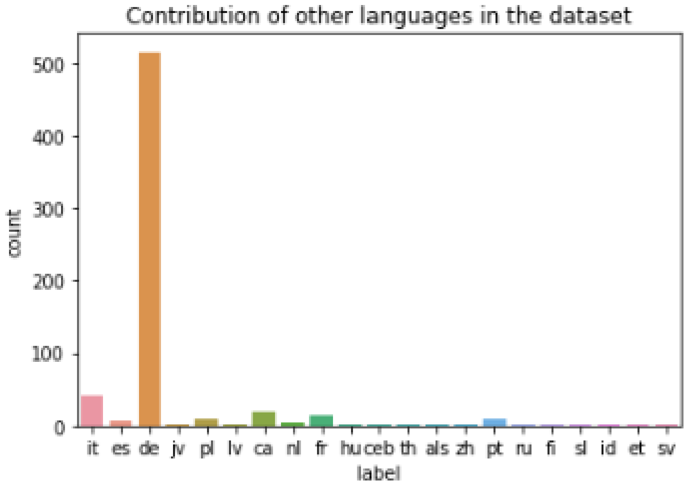


Figure 8 Contribution of other languages in the dataset

A total of 638 records are present for other languages. It is observed that 515 records belong to German language and the rest of 123 belong to other languages. Further detailing revealed that the model made mistakes of predicting other languages due to presence of special characters in them. The incidents belonging to languages other than English is just 7.5% of whole dataset. Hence we will discard these incidents from our dataset.

4. Deciding models and model building

Consider Figure 9 detailing the model building process that we have employed.

- First we have followed all data pre-processing steps as mentioned in Section 3.
- After that we have merged all columns into a single column and then ran the tokenizer on the merged column. The output of this step is provided to the word embedding step.
- The word embedding step uses GloVe embedding's for each word in the dataset.
- After this the actual skeleton of the model is created.

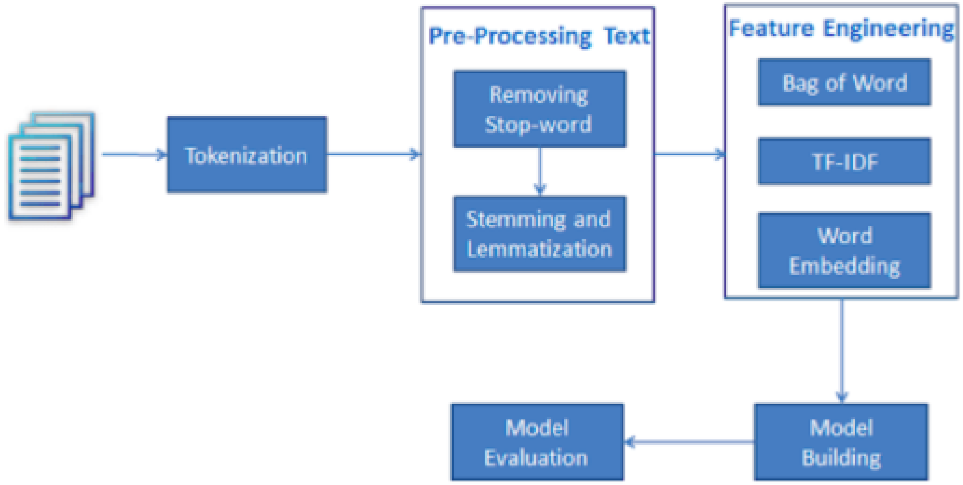


Figure 9 Model building process

4.1. Options for NLP classification model

There are three kind of models for NLP classification viz.

- **RNN** - is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Using the knowledge from an external embedding can enhance the precision of an RNN because it integrates new information (lexical and semantic) about the words, an information that has been trained and distilled on a very large corpus of data. The pre-trained embedding we can use for exampl is GloVe.

- **LSTM** - are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems, and are now widely used. STMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

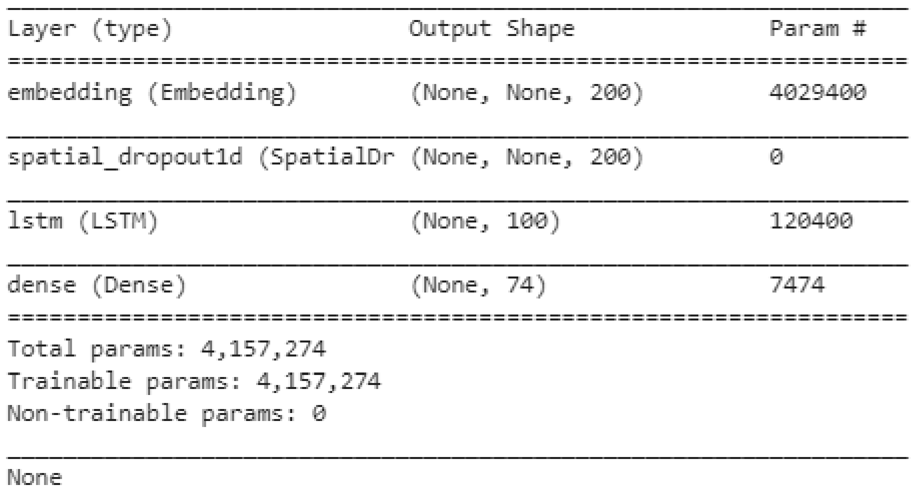
- **BERT** - is a technique for NLP pre-training developed by Google. BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text.

4.2. Model selection

We have used a Sequential **LSTM** model with **GloVe** embeddings. It has an input embedding layer, followed by spatial dropout layer.

The last layer is a fully connected layer with 74 outputs.

We have used the standard **adam** optimizer with the **categorical_crossentropy** as the measure of loss. Here is the model summary –



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 200)         4029400
_________________________________________________________________
spatial_dropout1d (SpatialDr (None, None, 200)         0
_________________________________________________________________
lstm (LSTM)                  (None, 100)               120400
_________________________________________________________________
dense (Dense)                (None, 74)                7474
=================================================================
Total params: 4,157,274
Trainable params: 4,157,274
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 10 Model summary

5. Model performance

The accuracy of the model on test set is 63.23%.

5.1. Improvements to model performance

Here are the points to be considered to improve the model performance-

- The data is highly imbalanced for Grp_0. The number of incidents corresponding to other groups need to be up-sampled.
- The total number of groups is high and some of the groups just have single entry in them. We can merge different groups together to improve accuracy.
- The BERT model can be tried to improve on model training time.

6. Summary

Most of data preprocessing steps are completed. The focus next is to work on improving the overall model performance. Various combination of models can be tried to achieve good results.