Automatic Ticket Assignment - FINAL REPORT

1. Summary of the problem statement

In the support process, incoming incidents are analysed and assessed by the organisation's support teams to fulfil the request. In many organisations, better allocation and effective usage of the valuable support resources will directly result in substantial cost savings. Using AI techniques, the incidents can be classified and assigned to the right functional groups.

In recent years, there has been exponential progress in the area of deep learning and machine learning methods to be able to accurately classify texts in many applications. Many Deep learning approaches have achieved surpassing results in natural language processing. The success of these learning algorithms relies on their capacity to understand complex models and non-linear relationships within data. However, finding suitable structures, architectures, and techniques for text classification is a challenge. In this report, a brief overview of text classification algorithms is discussed. This overview covers different text feature extractions and pre-processing various algorithms and techniques. Finally, the limitations of each technique and suggestions are discussed.

2. Overview of the Process

A. Problem Statement

Incidents are generally interruptions in the normal process which must be reported. These are reported by generating tickets. The organization has different groups to solve different types of issues. Therefore, based upon the ticket content we need to categories which group should be assigned the ticket.

B. Salient Features of Data

Following features are extracted from Data using various EDA techniques

- The provided dataset has 8500 samples with 74 different Groups to be classified.
- 46% of the samples are only from Group 0.
- There are almost 15 Groups, where not more than 3 samples are available; which are later categorized as 'Others'.
- There are only 9 records having NULL in the dataset. As the number is only 0.1% of total samples, hence we dropped those records.
- There are 2862 samples, where the content of short and long descriptions is exactly the same.
- 1828 people have reported their own complaint.
- Dataset has been expanded by creating new attributes like word counts, special character counts, stopwords counts, emails counts, date counts, image counts and reference counts.
- Using spacy_langdetect method, we came to know that almost 20 languages were used to report the incidents.
- Top 5 languages are English, German, Afrikaans, Italian and French.
- But the English language has 5283 out of 8500 samples, which is around 62% of the complete dataset.
- To limit the number of words for the models, we have used np.percentile method. It shows, 90 percentile of the Description field word count is 55 and 13 for Short Description respectively.
- Using NLTK, FreqDist, we came to know there are around 27605 distinct words present in the whole corpus.

C. Data Pre-Processing

We have developed a pre-processor helper Class, which has methods to remove unwanted words which may not help in classification.

- Stopwords and Custom words
- Usernames and Initials of names
- Html tags
- Emails
- Contractions or shortened version of a word
- Special characters
- Stemming the words
- Lemmatized the words

code location: src/dataset_preprocessor.py

D. Algorithms

We have tried following Neural networks to compare the classification results

- LSTM
- Bi-directional LSTM
- Bi-directional GRU
- Text CNN

E. Combined Techniques

With the above models, we have used various embedding techniques to get the optimum context of the words represented in vectors.

- GloVe
- Regularizer
- Elmo
- Attention
- BERT

- Single Channel

- Multi-Channel

3. Step-by-step walkthrough the solution

- Started with mining the data using various EDA techniques and analyzing the findings. Please refer section 2.B for details on findings.
- Based on EDA findings, we developed a pre-processor helper class to remove the unwanted words from the short and long description. Refer section 2.C
- The cleaned dataset has additional fields to show clean corpus along with a language identifier field and a new target column.
- New target field has a new group called ‘OTHER’ which is a consolidated category of groups where the sample size is not more than 3.
- Here onwards we are using incidents which are reported in English and discarded the samples reported in other languages.
- Next, convert the sentences to tokens using word_tokenize.
- Convert the words into sequences of numbers. These numbers are nothing but the word_counts of that word.
- Based on it, we converted the sentences into an array of numbers.
- Then divided the dataset into training and test sets into 80:20 ratio.
- Then started implementing various models. Refer section 2.D
- Based on various model accuracy, started implementing various embedding techniques. Refer section 2.E.

4. Model evaluation

We have selected Bi-directional GRU with simple attention as the final model. As this model gave us the best training accuracy

A. Bi-directional GRU with simple attention

- We chose Bi-directional GRU to derive proper context by looking at words in previous and post time steps
- Attention mechanism to pay more emphasis on relevant words so that the context is preserved in lengthy sentences
- There are several attention models - Simple and Hierarchical network attention (When the text has multiple sentences or when it's lengthy). In HAN's, we look at word level as well as Sentence level attention
- We used a pre-trained embedding layer (Glove) for the word embeddings

B. Objective

Our objective was to classify the reported incident into a specific group. By using the above model, we learnt the training weights and then finally pass through a softmax layer to give us the probability values for each of the 55 groups

C. Hyper-Parameters

- The pre-trained Glove embeddings were used as the word embeddings. We have two bi-directional layers stacked over each other, followed by an attention layer, dense layer and another dense layer with a softmax activation. We ran the model with various batch sizes, epochs along with multiple dense layers with activation function like "RELU".

D. Evaluation

- We fit the model with a training set which is 80% of the total sample (only English).
- During the fit, we used 20% of the training set as a validation set.
- We obtained a training accuracy of 81.1 and a testing accuracy of 61.13 (batch size: 32, Epochs: 15)

5. Comparison to benchmark

We have used the result of Bi-directional LSTM recurrent network as a benchmark.

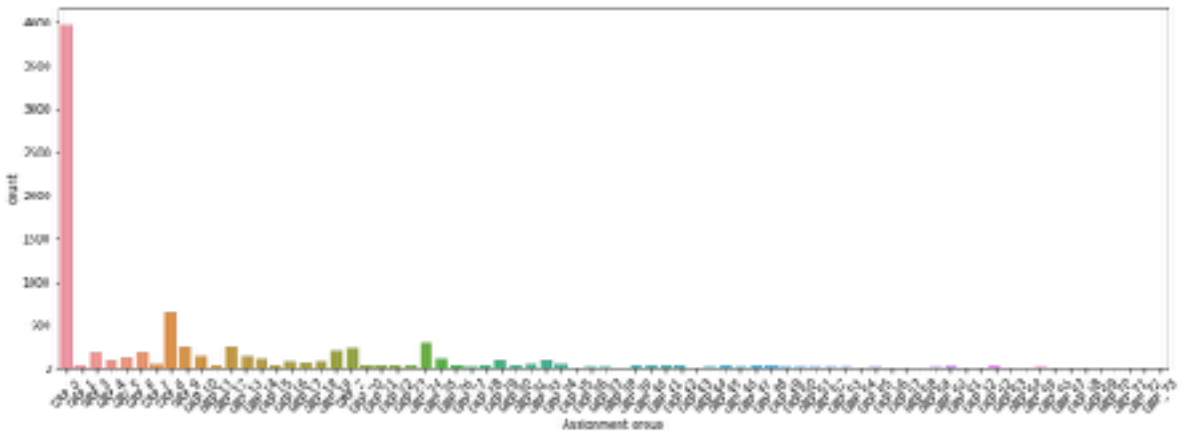
Repo Link: https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/tree/master

Models	# Epochs	Test_ Accuracy (%)	F1 Score	File Name
LSTM	10	55.15	0.55	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/models/LSTM.ipynb
Regularizers+Bi-dir LSTM	10	58.54	0.59	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/models/LSTM.ipynb
Glove+Bi-dir LSTM	10	57.44	0.75	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/Capstone_Project_v3.0.ipynb
ELMO	20	43.98	0.68	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/Capstone_Project_v3.0.ipynb
Hierarchical Attention+LSTM	10	55.82	0.78	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/Capstone_Project_v3.0.ipynb
Single Channel text CNN	25	51.18	0.75	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/Capstone_Project_v3.0.ipynb
Multi-Channel Text CNN	10	45.87	0.63	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/Capstone_Project_v3.0.ipynb
GRU	5	55.99	0.56	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/models/GRU.ipynb
Regularizers+Bi-dir GRU	5	57.22	0.57	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/models/bidirectional_GRU.ipynb
Glove+Bi-dir GRU	5	56.09	0.56	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/models/glove_bidirectional_GRU.ipynb
Attention+Bi-dir GRU(Final Model)	15	61.13	0.80	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/Capstone_Project_v3.0.ipynb
BERT	2	47.93	NA	https://gitlab.com/vinay.katdare/capstone_aimlmarchgroup2_b_nlp/-/blob/inital_eda/models/Bidir_LSTM_ELMO_Attention_BERT_Models_v1.ipynb

6. Visualizations

Following are the sample visualizations we extracted out of the EDA process.

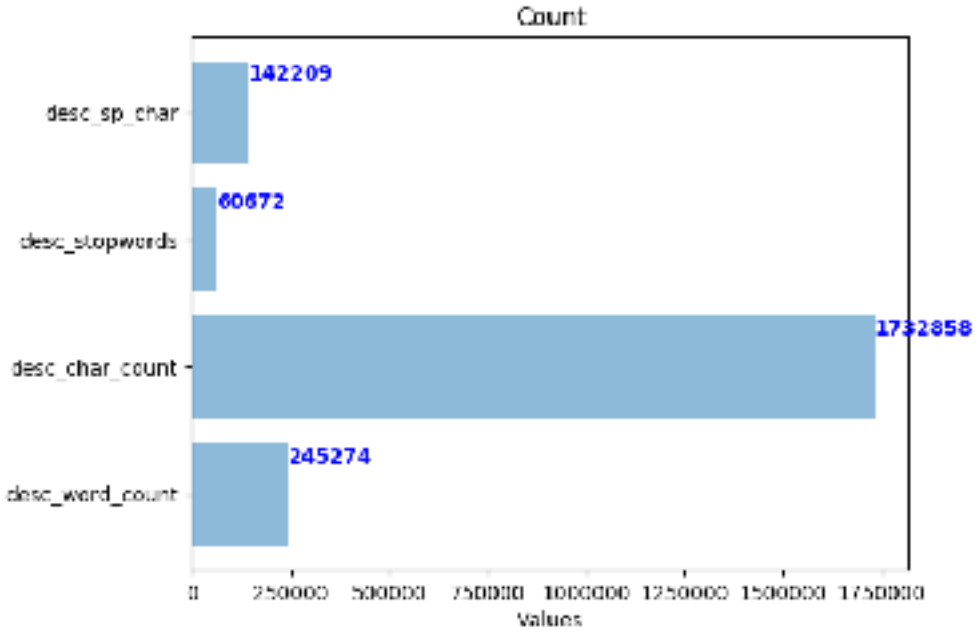
- Distribution of Target field



- Data having Null values

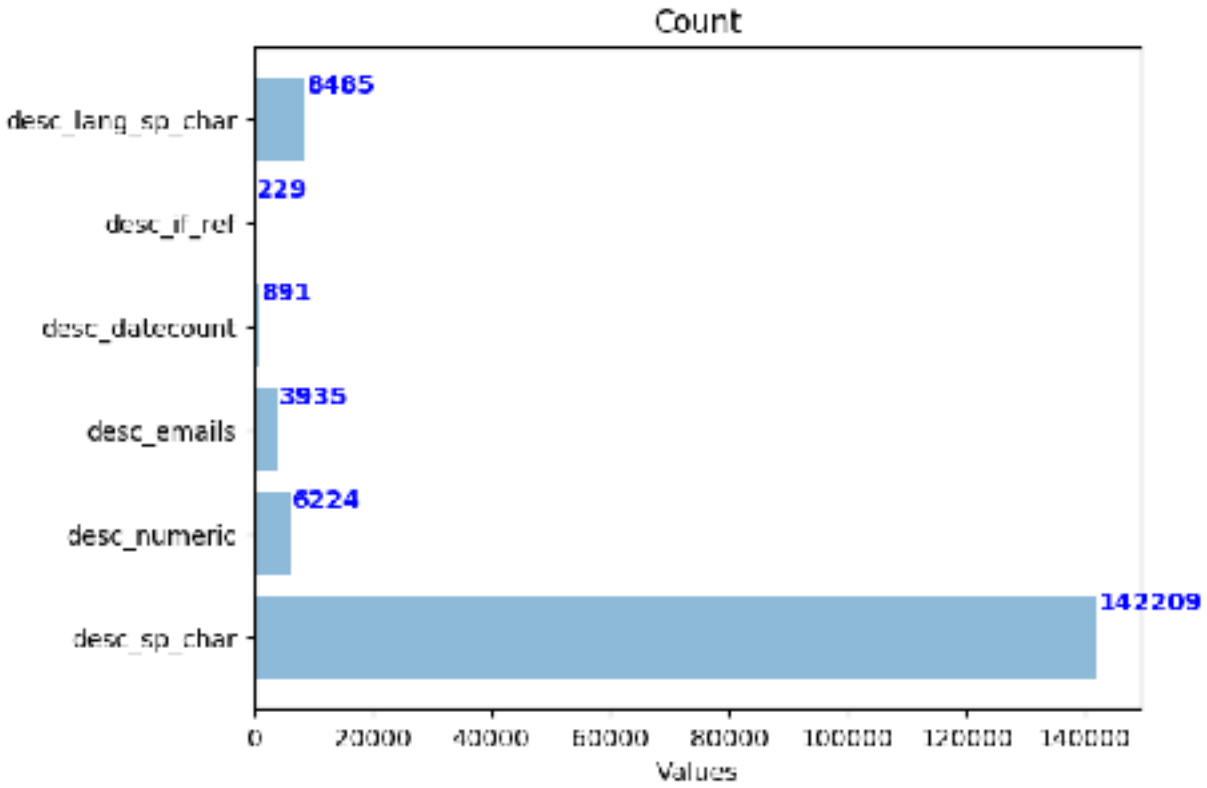
	Short description	Description	Caller	Assignment group
2604	NaN	\r\n\r\nreceived from: ohdrnswl.rezuibdt@gmail...	ohdrnswl rezuibdt	GRP_34
3383	NaN	\r\n-connected to the user system using teamvi...	qftpazns fxpnytmk	GRP_0
3906	NaN	-user unable tologin to vpn.\r\n-connected to...	awpcmsey ctdluqwe	GRP_0
3910	NaN	-user unable tologin to vpn.\r\n-connected to...	rhwsmefo tvphyura	GRP_0
3915	NaN	-user unable tologin to vpn.\r\n-connected to...	hxripljo efzounig	GRP_0
3921	NaN	-user unable tologin to vpn.\r\n-connected to...	cziadygo veiosxby	GRP_0
3924	NaN	name:wvqgbdhm fwchqjor\nlanguage:\nbrowser:mic...	wvqgbdhm fwchqjor	GRP_0
4341	NaN	\r\n\r\nreceived from: eqmuniov.ehxkcbgj@gmail...	eqmuniov ehxkcbgj	GRP_0
4395	I am locked out of skype	NaN	viyglzfo ajtfzpkb	GRP_0

- Bar chart showing the comparative counts of stopwords, characters, words and special

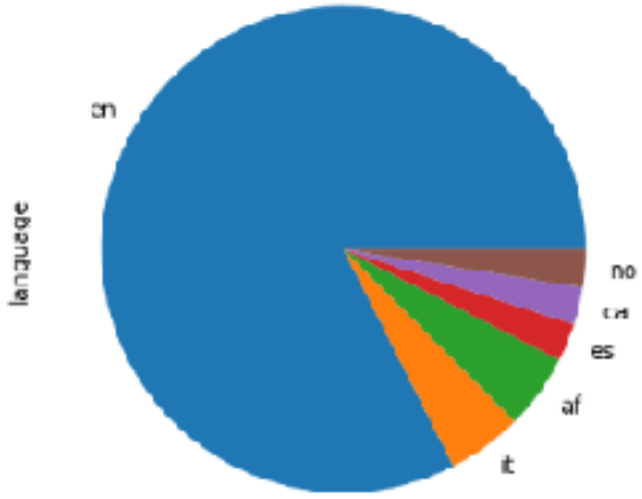


character

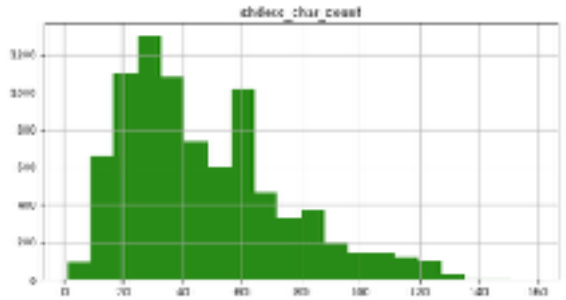
- Bar chart of the total count of special characters, numbers, emails, date count, hyperlinks, language

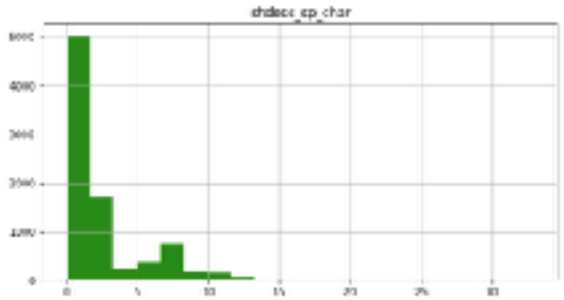


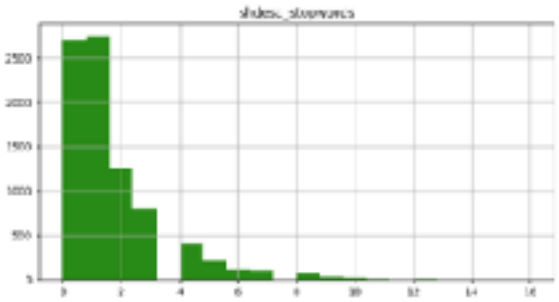
- Pie chart with respect to the distribution of language words throughout the text

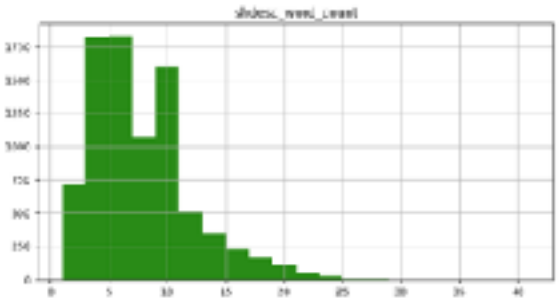


- Histogram of distribution of word count, character count, stopwords, and special characters

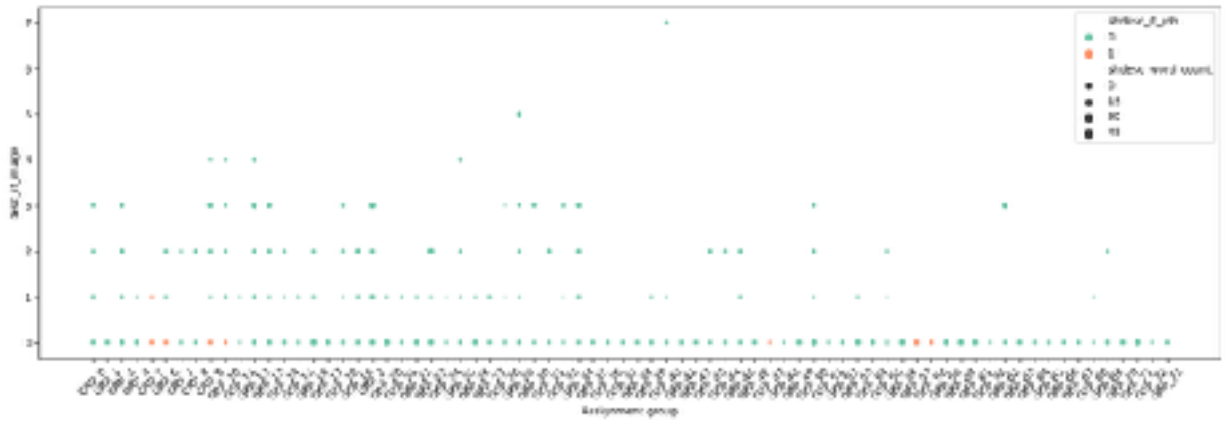








- Scatter plot to show if Image references have been given in tickets where job/ hostname/ticket numbers are mentioned



7. Implications

- The solution is going to enhance the serviceability of both internal and external customers.
- It removes the time delays introduced by manual segregation and allocation of tickets to the appropriate groups.
- Considering the volumes of tickets, it's very likely to create a backlog for allocation to appropriate groups and also introduces delay by allocating to a wrong group.
- A mature ML solution like this is very likely to remove all such issues and enhances the experience of the end customer for a timely resolution
- This solution can also be modified for different organizations according to their data

8. Limitations

Dataset is very biased. Of the 5271 English records & 55 groups, 50% of the records belong to the major group and the remaining 54 groups account for the remaining 50%.On average, each of the remaining 54 groups accounts for 5% of the records.

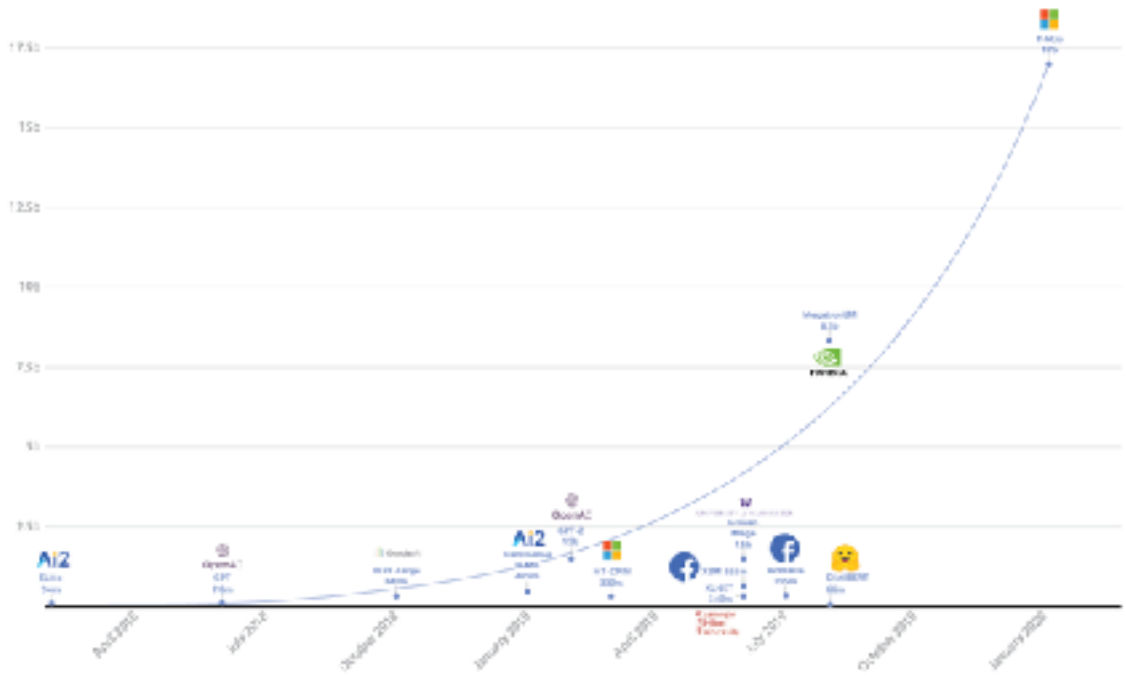
- Collect more data from the source and with uniform variation
- Data augmentation using upsampling or downsampling techniques

9. Closing Reflections

The problem at hand has been an enormous learning experience. We had to address several challenges of cleaning the dataset, deciding on our approach for handling the multilingual data. Given the size of the training data, we had to try out different modelling techniques to get the best accuracy.

Approach:
We wanted to try out different NLP libraries so that we get a sense of the vastness of NLP and also the latest NLP libraries used in the market.

The following image shows all the NLP models in the market along with its timeline-



We have successfully done that and implemented close to 12 libraries and found a lot about the pros and cons of each library.

Following are the libraries that we have implemented:

1. LSTM
2. Regularizers+Bi-dir LSTM
3. Glove+Bi-dir LSTM
4. ELMO
5. Hierarchical Attention+LSTM
6. Single Channel text CNN
7. Multi-Channel Text CNN
8. GRU
9. Regularizers+Bi-dir GRU
10. Glove+Bi-dir GRU
11. Attention+Bi-dir GRU(Final Model)
12. BERT

We've also tried Distil-Bert and Fast.ai Bert but couldn't get it to work on Google Colab as the Memory was being over-shot. We learnt something new here too.

Moving Forward:
We are working towards deploying this model on Heroku server and BUilding a sample website so that people can try out our prediction using different models as a dropdown and adding the text which would, in turn, hit our API and generate the results.

In conclusion, We are very happy that we approached the problem horizontally as planned instead of deep-diving into 1 single model.

Quote:

“Never get discouraged if you fail. Learn from it. Keep trying” - Thomas Edison

Thank You.