NLP CAPSTONE PROJECT – Interim Report

NLP-2 Semi Ruled ChatBot

Content

Overview

Domain - Industrial safety NLP based Chatbot

Context:
The database comes from one of the biggest industry in Brazil and in the world. It is an urgent need for industries/companies around the globe to understand why employees still suffer some injuries/accidents in plants. Sometimes they also die in such environment.

Data Description:
This The database is basically records of accidents from 12 different plants in 03 different countries which every line in the data is an occurrence of an accident.

Columns description:

Data: timestamp or time/date information

Countries: which country the accident occurred (anonymised)

Local: the city where the manufacturing plant is located (anonymised)

Industry sector: which sector the plant belongs to

Accident level: from I to VI, it registers how severe was the accident (I means not severe but VI means very severe)

Potential Accident Level: Depending on the Accident Level, the database also registers how severe the accident could have been (due to other factors involved in the accident)

Genre: if the person is male of female

Employee or Third Party: if the injured person is an employee or a third party

Critical Risk: some description of the risk involved in the accident

Description: Detailed description of how the accident happened.

2. Data Analysis

2.1 Data Collection

Raw Data :

	Unnamed: 0	Data	Countries	Local	Industry Sector	Accident Level	Potential Accident Level	Genre	Employee or Third Party	Critical Risk	Description
0	0	2016-01-01 00:00:00	Country_01	Local_01	Mining	I	IV	Male	Third Party	Pressed	While removing the drill rod of the Jumbo 08 f...
1	1	2016-01-02 00:00:00	Country_02	Local_02	Mining	I	IV	Male	Employee	Pressurized Systems	During the activation of a sodium sulphide pum...
2	2	2016-01-06 00:00:00	Country_01	Local_03	Mining	I	III	Male	Third Party (Remote)	Manual Tools	In the sub-station MILPO located at level +170...
3	3	2016-01-08 00:00:00	Country_01	Local_04	Mining	I	I	Male	Third Party	Others	Being 9:45 am. approximately in the Nv. 1880 C...
4	4	2016-01-10 00:00:00	Country_01	Local_04	Mining	IV	IV	Male	Third Party	Others	Approximately at 11:45 a.m. in circumstances t...

- There are about 425 rows and 11 columns in the dataset.
- We noticed that except a 'date' column all other columns are categorical columns.

2.2 Data Cleaning

- Removed 'Unnamed: 0' column and renamed - 'Data', 'Countries', 'Genre', 'Employee or Third Party' columns in the dataset.
- We had 7 duplicate instances in the dataset and dropped those duplicates.
- There are no outliers in the dataset.
- No missing values in dataset.
- We are left with 418 rows and 10 columns after data cleansing.

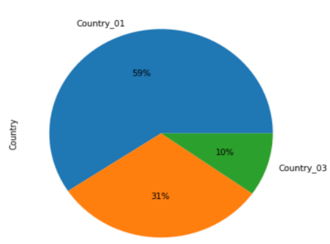
2.3 Exploratory Data Analys

2.3.1 Variable Identification:

- Target variable: 'Accident Level', 'Potential Accident Level'
- Predictors (Input varibles): 'Date', 'Country', 'Local', 'Industry Sector', 'Gender', 'Employee type', 'Critical Risk', 'Description'

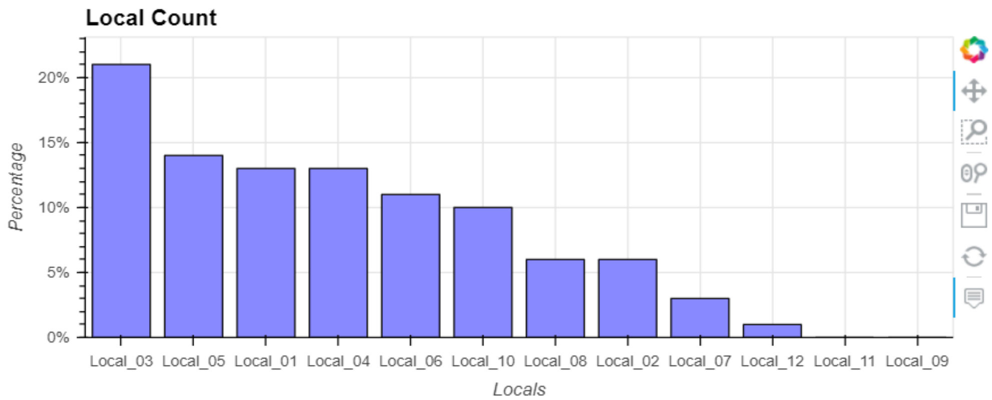
2.3.2 Univariate Analysis

- DATE: Here, the column name is renamed from Data to Date. Also, the year, month and day variables are extracted from the date column to find if there are any seasonality co-occurrences of accident.
- Country:



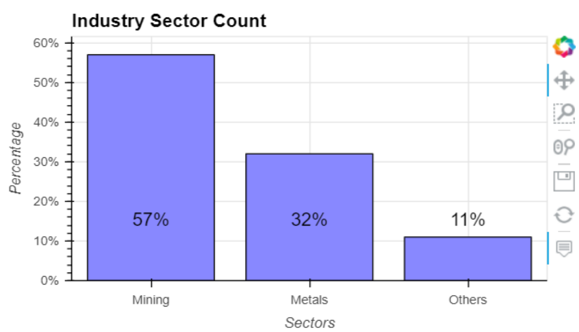
i. 59% accidents occurred in Country_01
ii. 31% accidents occurred in Country_02
iii. 10% accidents occurred in Country_03

- Local:



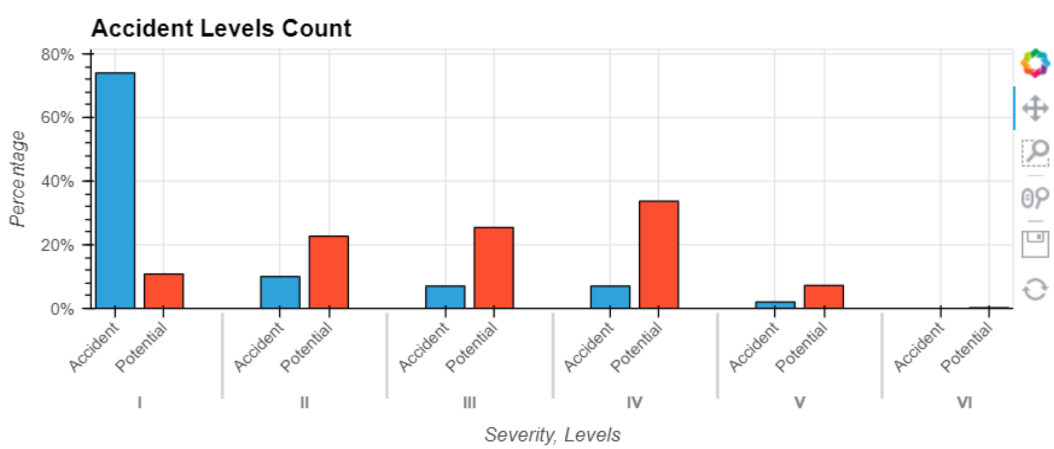
i. Highest manufacturing plants are located in Local_03 city.
ii. Lowest manufacturing plants are located in Local_09 city.

- Industry Sector:



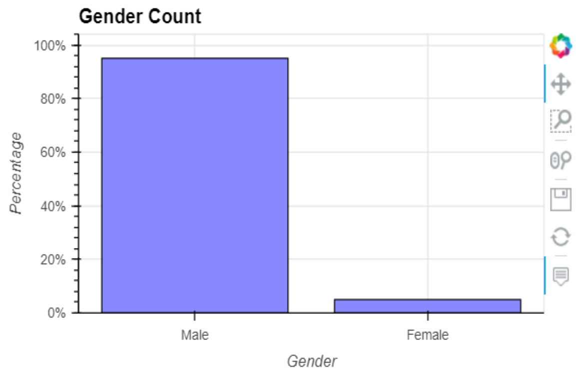
i. 57% manufacturing plants belongs to Mining sector.
ii. 32% manufacturing plants belongs to Metals sector.
iii. 11% manufacturing plants belongs to Others sector.

- Accident Levels:



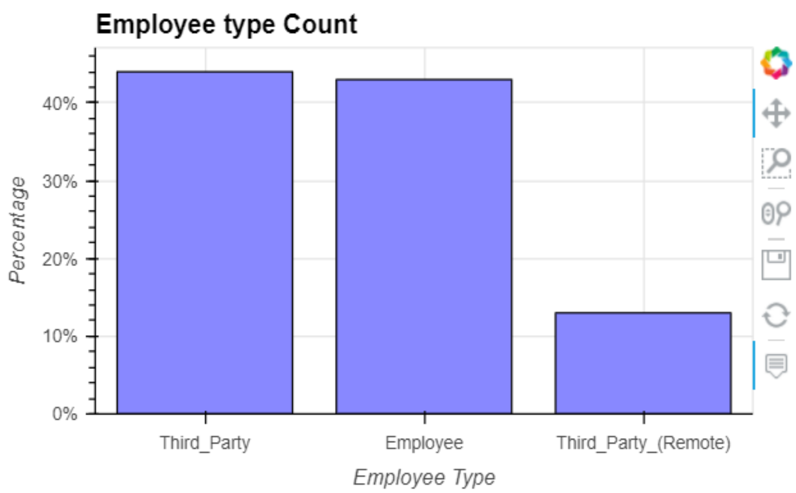
i. The number of accidents decreases as the Accident Level increases.
ii. The number of accidents increases as the Potential Accident Level increases.

- Gender:



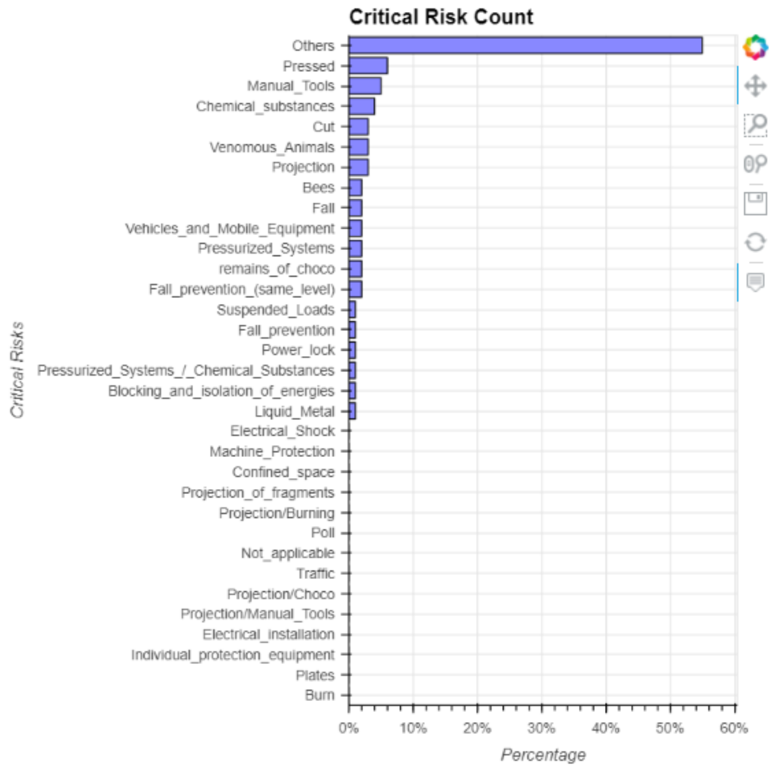
i. There are more men working in this industry as compared to women.

- Employee type:



i. 44% Third party employees, 43% own employees and 13% Third party(Remote) employees working in this industry.

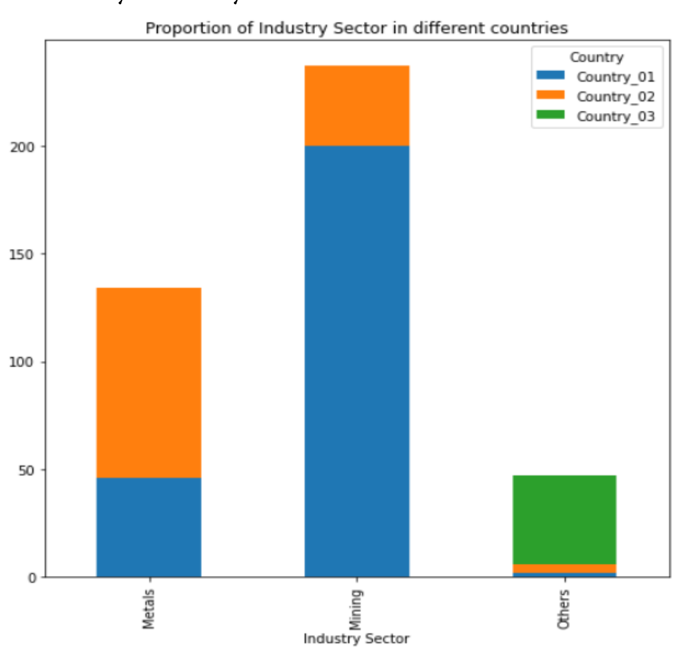
- Critical Risk:



i. Most of the incidents are registered as 'Others', it takes lot of time to analyze risks and reasons why the accidents occur.

2.3.3 Bivariate Analysis

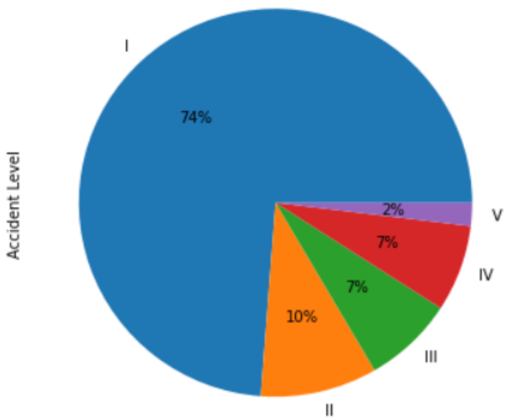
- Industry Sector by Countries:



i. Metals and Mining industry sector plants are not available in Country_03.
ii. Distribution of industry sector differ significantly in each country.

3. NLP Analysis

- Distributon of accident_level where the length of Description is greater than 100



i. 74% of data where accident description > 100 is captured in low accident level.
ii. Based on some random headlines seen above, it appears that the data is mostly lower-cased. Pre-processing such as removing punctuations and lemmatization can be used.
iii. There are few alphanumeric characters like 042-TC-06, Nv. 3370, CX 212 captured in description where removing these characters might help.
iv. There are digits in the description for e.g. level 326, Dumper 01 where removing the digits wouldn't help.

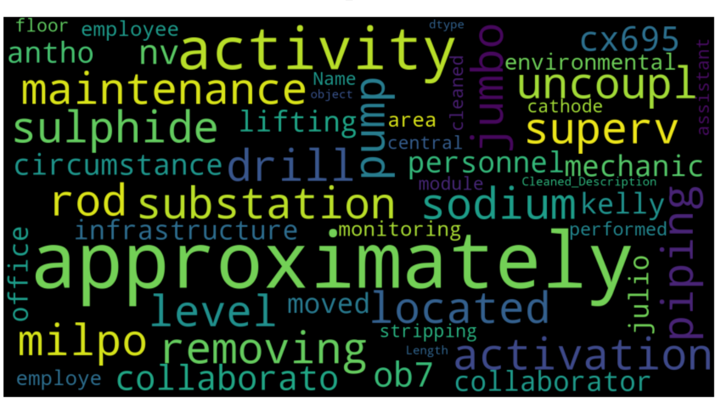
4. NLP Pre-processing

- Few of the NLP pre-processing steps taken before applying model on the data

i. Converting to lower case, avoid any capital cases
i. Converting apostrophe to the standard lexicons
ii. Removing punctuations
iii. Lemmatization
iv. Removing stop words

- Wordcloud

Wordcloud for cleaned description:



i. Most words are related to maintenance, accident, employee, equipment, infrastructure.

5. Data Modelling

Long-Short Term Memory (LSTM) with Glove embedding:
LSTM is a type of Recurrent Neural Network in Deep Learning that has been specifically developed for the use of handling sequential prediction problems. For example: Weather Forecasting, Stock Market Prediction, Product Recommendation, Text/Image/Handwriting Generation, Text Translation

GLOVE Embedding:
GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Glove file Used:glove.6B.300d.txt

Model Summary:

```
lstm_model.summary()

Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 32, 300)           4236000

 lstm_1 (LSTM)               (None, 64)                93440

 dense_1 (Dense)             (None, 14120)             917800

=================================================================
Total params: 5,247,240
Trainable params: 1,011,240
Non-trainable params: 4,236,000
_________________________________________________________________
```

Model Performance:

```
[86] history = lstm_model.fit(X_train, y_train, epochs=4, batch_size=1024, validation_data=(X_test, y_test))

Epoch 1/4
1/1 [==============================] - 6s 6s/step - loss: 9.5577 - accuracy: 0.0000e+00 - val_loss: 9.5244 - val_accuracy: 0.0312
Epoch 2/4
1/1 [==============================] - 0s 63ms/step - loss: 9.5263 - accuracy: 0.0101 - val_loss: 9.4884 - val_accuracy: 0.7109
Epoch 3/4
1/1 [==============================] - 0s 68ms/step - loss: 9.4932 - accuracy: 0.6195 - val_loss: 9.4475 - val_accuracy: 0.7500
Epoch 4/4
1/1 [==============================] - 0s 61ms/step - loss: 9.4551 - accuracy: 0.7273 - val_loss: 9.4001 - val_accuracy: 0.7656
```

Classification Report:

```
[87] print(classification_report(y_test, np.argmax(lstm_model.predict(X_test), axis=-1)))

              precision    recall  f1-score   support

           1       0.77      1.00      0.87        98
           2       0.00      0.00      0.00        11
           3       0.00      0.00      0.00         8
           4       0.00      0.00      0.00         9
           5       0.00      0.00      0.00         2

    accuracy                           0.77       128
   macro avg       0.15      0.20      0.17       128
weighted avg       0.59      0.77      0.66       128
```

BILSTM with glove Embeddings:

Bidirectional long-short term memory(bi-lstm) is the process of making any neural network o have the sequence information in both directions backwards (future to past) or forward(past to future). In bidirectional, our input flows in two directions, making a bi-lstm different from the regular LSTM. With the regular LSTM, we can make input flow in one direction, either backwards or forward. However, in bi-directional, we can make the input flow in both directions to preserve the future and the past information BiLSTM Summary:

Glove file Used:glove.6B.300d.txt

Model Summary:

```
[100] biLSTM.summary()

Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 32, 300)           4236000

 bidirectional_2 (Bidirectio  (None, 32, 128)          186880
 nal)

 bidirectional_3 (Bidirectio  (None, 64)               41216
 nal)

 dense_3 (Dense)             (None, 14120)             917800

=================================================================
Total params: 5,381,896
Trainable params: 1,145,896
Non-trainable params: 4,236,000
_________________________________________________________________
```

Model Performance:

```
[86] history = lstm_model.fit(X_train, y_train, epochs=4, batch_size=1024, validation_data=(X_test, y_test))

Epoch 1/4
1/1 [==============================] - 6s 6s/step - loss: 9.5577 - accuracy: 0.0000e+00 - val_loss: 9.5244 - val_accuracy: 0.0312
Epoch 2/4
1/1 [==============================] - 0s 63ms/step - loss: 9.5263 - accuracy: 0.0101 - val_loss: 9.4884 - val_accuracy: 0.7109
Epoch 3/4
1/1 [==============================] - 0s 68ms/step - loss: 9.4932 - accuracy: 0.6195 - val_loss: 9.4475 - val_accuracy: 0.7500
Epoch 4/4
1/1 [==============================] - 0s 61ms/step - loss: 9.4551 - accuracy: 0.7273 - val_loss: 9.4001 - val_accuracy: 0.7656
```

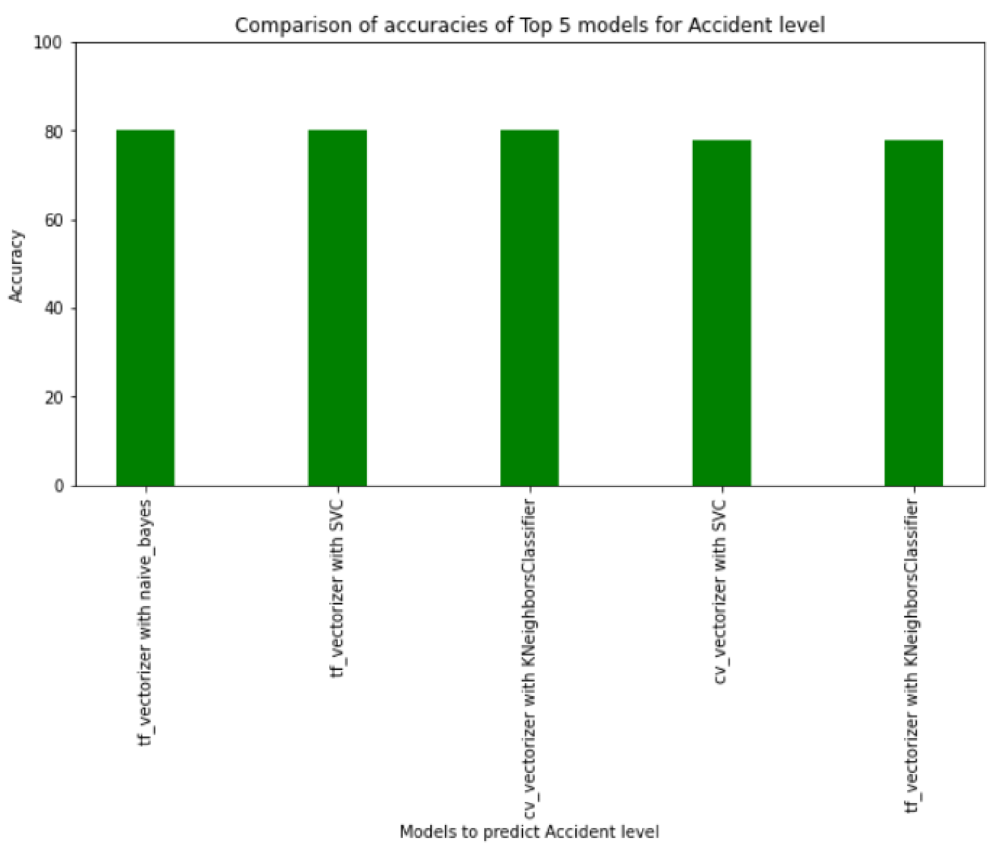
Classification Report:

```
] print(classification_report(y_test, np.argmax(biLSTM.predict(X_test), axis=-1)))

              precision    recall  f1-score   support

           1       0.77      1.00      0.87        98
           2       0.00      0.00      0.00        11
           3       0.00      0.00      0.00         8
           4       0.00      0.00      0.00         9
           5       0.00      0.00      0.00         2

    accuracy                           0.77       128
   macro avg       0.15      0.20      0.17       128
weighted avg       0.59      0.77      0.66       128
```

The following table displays all other models performance:

Model	Accuracy
Random Forest Classifier with Glove Accuracy	0.72
Bagging Classifier with Glove Accuracy	0.69
TF-IDF with Naïve bayes	0.80
TF-IDF with SVC	0.80
TF-IDF with KNeighborsClassifier	0.78
Countvectorizer with Naïve bayes	0.76
Countvectorizer with SVC	0.78
Countvectorizer with KNeighborsClassifier	0.80

Model Comparison:



In comparison of all the above models for target label Accident level, we can say that tf_vectorizer with naive_bayes , tf_vectorizer with SVC and cv_vectorizer with KNeighborsClassifier shows similar accuracy.

6. How to improve Model performance?

- Enhancements can be made to improve performance by Model tuning and using sampling techniques like SMOTE for imbalanced data.
- By changing parameters of the model using other vectorizer techniques like ELMO etc.
- Data Augmentation methods can be used to improve performance.