Paper ID- IRCICD_2023_22

“Interpretability of a Joint Learning Problem”

By Abdul Mannan, Ishita Chaudhary, Rithika Shankar, Bhaskarjyoti Das

Contents

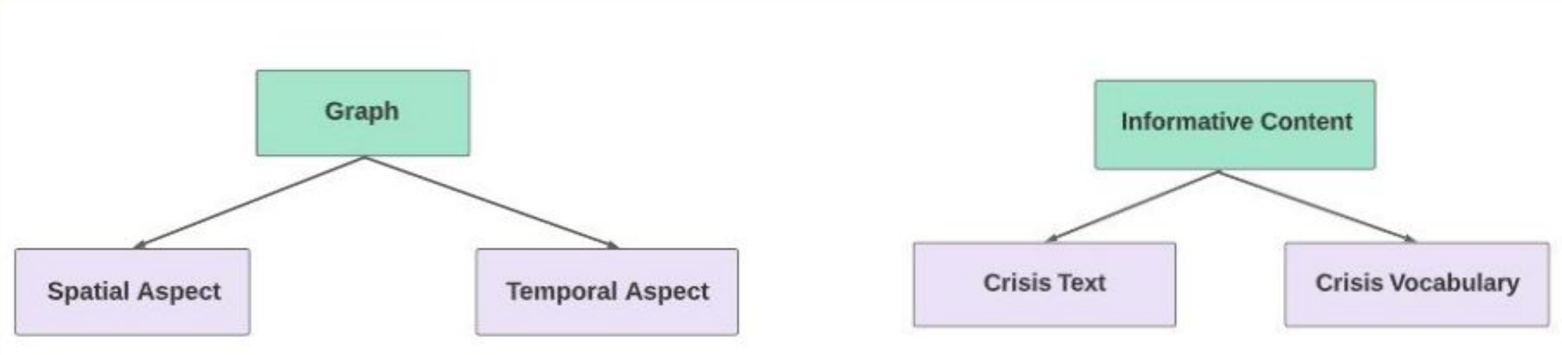
Problem Statement

"Interpretability of a joint learning problem by utilising a social graph as well as by understanding the context of the textual data”

Introduction

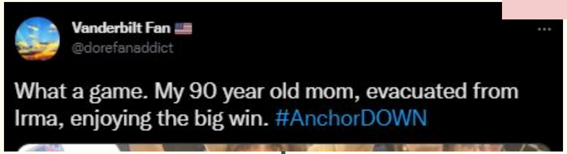
Identifying whether Twitter can be used as a reliable source during a natural calamity by utilizing the social graph as well as by understanding the text

We then look into the interpretability of this model.



Motivation

Informative



Not Informative

Joint Model: Not Informative

```
Original Text: What a game. My 90 year old mom, evacuated from Irma, enjoying the big win. #AnchorDOWN
True Label: 1
Predicted Label by joint model: 1
Predicted Label by text model: 0
Predicted Label by graph model: 1
```

Literature Survey

No.	Paper Title	Author(s)	Focus Area	Publication
1.	A Twitter Tale of Three Hurricanes: Harvey, Irma, and Maria	Firoj Alam, Ferda Ofli, Muhammad Imran, Michael Aupetit	Multi-modal analysis of a crisis using text and images	15th International Conference on Information Systems for Crisis Response and Management (ISCRAM), 2018.
2.	A Neural-Based Approach for Detecting the Situational Information From Twitter During Disaster	Sreenivasulu Madichetty, Sridevi M	Superiority of deep learning models over traditional algorithms in text classification	IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 8, NO. 4, AUGUST 2021
3.	Analysis of Community Response to Disasters through Twitter Social Media	Apri Junaidi, Iqsyahiro Kresna, Richki Hardi	Analysis of user activity in response to a disaster on twitter	ICSINTESA 2019 Journal of Physics: Conference Series
4.	A Fake News Detection Framework Using Social UserGraph	Yi Xie, Xixuan Huang, Xiaoxuan Xie	User profiling+use of GraphSage to improve detection of fake news	2020 Association for Computing Machinery

Literature Survey

No.	Paper Title	Author(s)	Focus Area	Publication
5.	Understanding communication dynamics on Twitter during natural disasters: A case study of Hurricane Sandy	Nastaran Pourebrahim, Selima Sultana, John Edwards, Amanda Gochanour, Somya Mohanty	A thorough analysis of the usage of twitter (temporal,textual,text,user sentiment,social network views) during a natural disaster	International Journal of Disaster Risk Reduction,Volume 37
6.	I-AID: Identifying Actionable Information from Disaster-related Tweets	Hamada M. Zahera, Rricha Jalota, Mohamed A. Sherif, Axel N. Ngomo	Using GAT for improving textual analysis and classification of tweets	IEEE Access
7.	Semi-supervised Stance Detection of Tweets Via Distant Network Supervision	Subhabrata Dutta, Samiya Caur, Soumen Chakrabarti, Tanmoy Chakraborty	Semi-supervised learning using social network for providing coarse user stance	WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining
8.	MVGAN: Multi-View Graph Attention Network for Social Event Detection	Yi Xie, Xixuan Huang, Xiaoxuan Xie	Use of heterogeneous graphs and GCN’s from a semantic and temporal view for event detection	ACM Transactions on Intelligent Systems and Technology,Volume 12,Issue 3

Key Takeaways from Literature Survey

1. Use of multiple views of data can greatly improve prediction of events and classification of tweets.

2. Use of temporal information and geo-tagging of tweets helps in detecting credible tweets in a disaster.

3. Ignoring the conversation structure can lead to information misinterpretation and affect accuracy of classification

Key Takeaways from Literature Survey

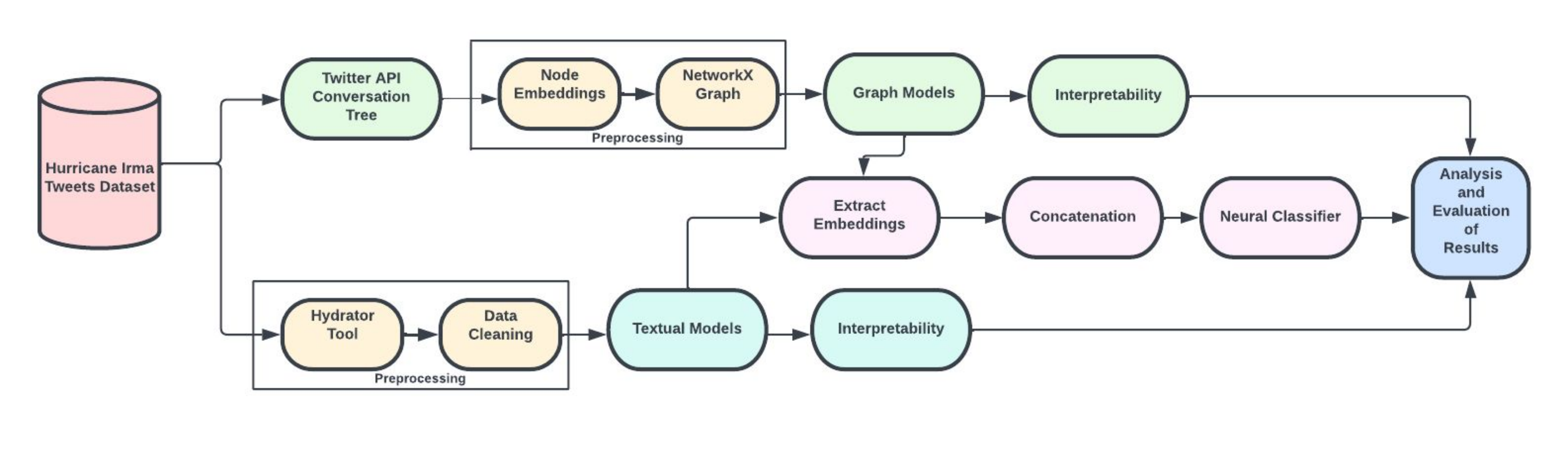
4. A Graph attention network is useful to capture the semantics between words and entities, can be used to capture spatial and temporal aspects too.

5. Multi-heads attention makes the attention network more robust, as each attention network can capture different connections between nodes.

6. Co-attention schemes can help improve classification accuracy + provide explainability

Methodology

The model architecture is as follows



Data Collection & Preparation

To prepare our dataset, we used a Hydrator tool to extract additional tweet data.

```
In [5]: list(f)

Out[5]: ['tweet_id',
         'image_id',
         'text_info',
         'text_info_conf',
         'image_info',
         'image_info_conf',
         'text_human',
         'text_human_conf',
         'image_human',
         'image_human_conf',
         'image_damage',
         'image_damage_conf',
         'tweet_text',
         'image_url',
         'image_path']
```

```
In [8]: list(df2)

Out[8]: ['coordinates',
         'created_at',
         'hashtags',
         'media',
         'urls',
         'favorite_count',
         'id',
         'in_reply_to_screen_name',
         'in_reply_to_status_id',
         'in_reply_to_user_id',
         'lang',
         'place',
         'possibly_sensitive',
         'quote_id',
         'retweet_count',
         'retweet_id',
         'retweet_screen_name',
         'source',
         'text',
         'tweet_url',
         'user_created_at',
         'user_id',
         'user_default_profile_image',
         'user_description',
         'user_favourites_count',
         'user_followers_count',
         'user_friends_count',
         'user_listed_count',
         'user_location',
         'user_name',
         'user_screen_name',
         'user_statuses_count',
         'user_time_zone',
         'user_urls',
         'user_verified']
```

Data Pre-Processing

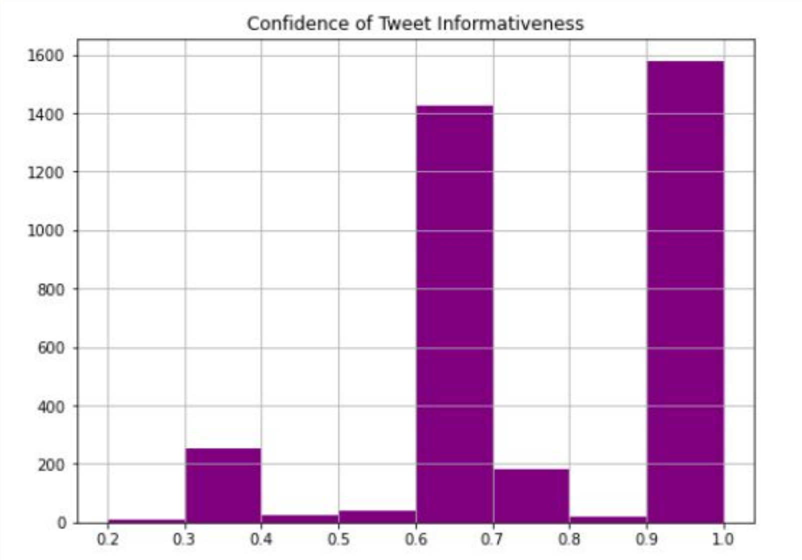
To improve our textual model, we did the following steps to clean our data

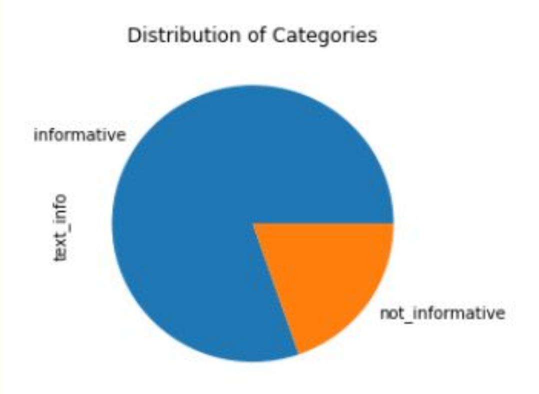
```
In [26]: df['text']

Out[26]: 0       Island of Barbuda 'literally under water' afte...
         1       JUST IN: 11PM #Hurricane #Irma update. @ABC7Ne...
         2       Hurricane Irma destroys "upwards of 90%" of Ba...
         3       5 PM track and update for Hurricane Irma. #flw...
         4       Here is the 11 pm advisory for Hurricane #Irma...
                                        ...
         3515    Hurricane Irma First Impacts On Nassau Bahamas...
         3516        Local Resources #Irma https://t.co/CphnkQ6y37
         3517    Whoa! Is @Delta about to pull off one more las...
         3518    New story in Science &amp; Health from Time: h...
         3519    Watch: Two cyclists spotted out for a ride in ...
         Name: text, Length: 3520, dtype: object
```

```
In [14]: df['text']

Out[14]: 0       email email wh respond faster incid london ear...
         1       least eight dead includ toddler num injur hurr...
         2       watch live gov nathan deal survey damag forom ...
         3       email quick tweet h tag sympathi mexico earthq...
         4       email email health group donat num million flo...
                                        ...
         6743    hour grey phal hayl oyster bed w hayl lnr floo...
         6744    made h tag wy damag surround fenc tree okay ot...
         6745    tebow share moment wwii veteran hurrican irma ...
         6746    h tag morton salt say hurrican irma damag baha...
         6747                   thought prayer path h tag dig h tag
         Name: text, Length: 6748, dtype: object
```

Data Visualization

To understand the distribution of tweets in our dataset, visualized the proportion of tweets in each category





Textual Model- 1

Embedding-bag classifier with LIME for interpretability

Test Accuracy	80.50

```
print('Prediction probability:', round(probs[test_labels[0]].item(), 4))
```

```
Prediction probability: 0.9303
```

```
show_text_attr(attrs)
```

dispatch respond area hit hard h tag vonda doreen napl assist victim hurrican irma proud ! h tag

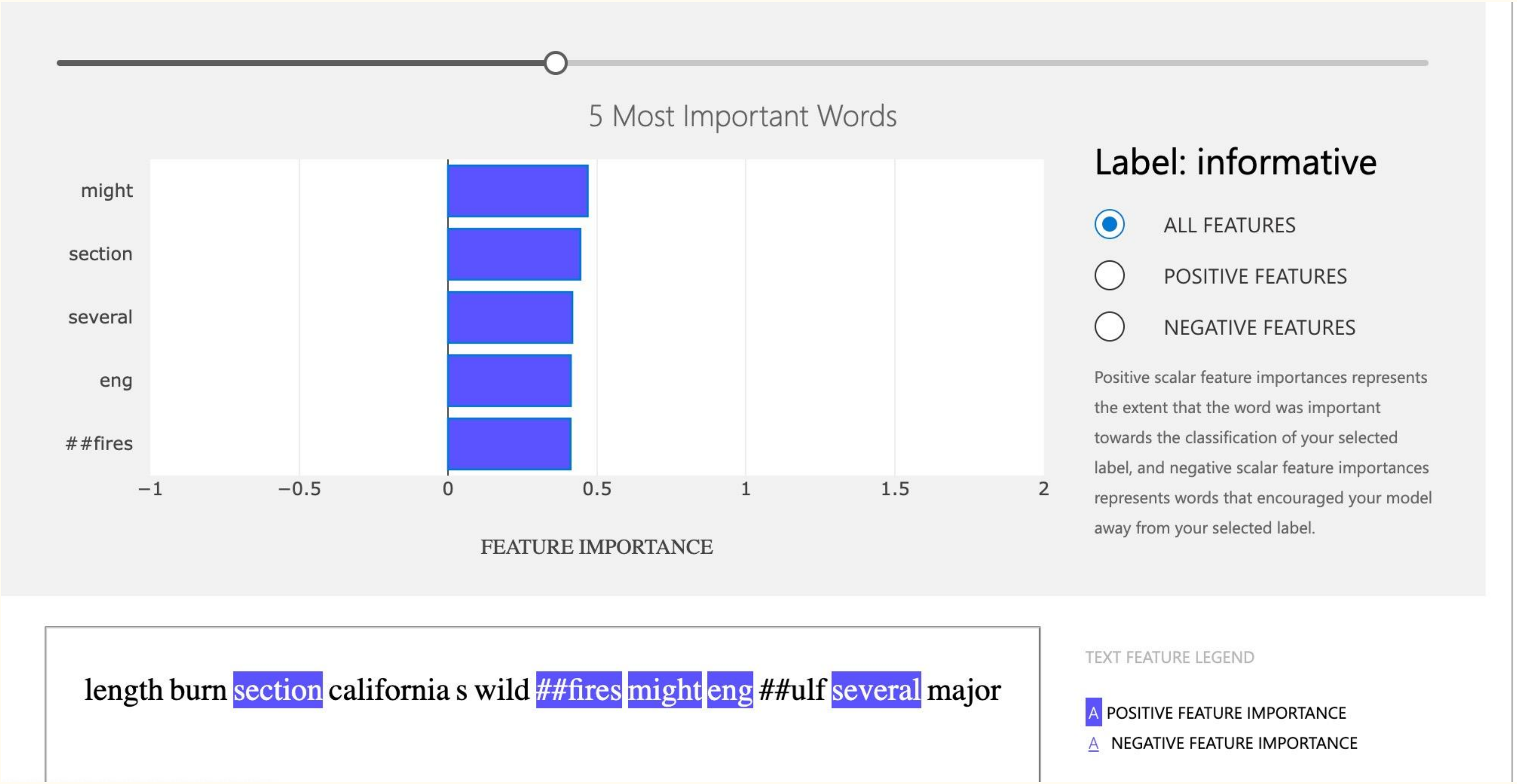
Textual Model- 2

BertSequenceClassifier with interpret-text for interpretability

- We use Matthews correlation coefficient as a metric since it provides a more reliable result.
- The Matthews correlation coefficient (MCC) produces a high score only if the prediction obtained good results in all of the four confusion matrix categories.
- It measures the level of agreement or disagreement between the predictions and true labels.
- The value of MCC can range between -1 to 1 where 0 being random guess, -1 being complete disagreement and +1 being complete agreement

Test Accuracy	**85.00**

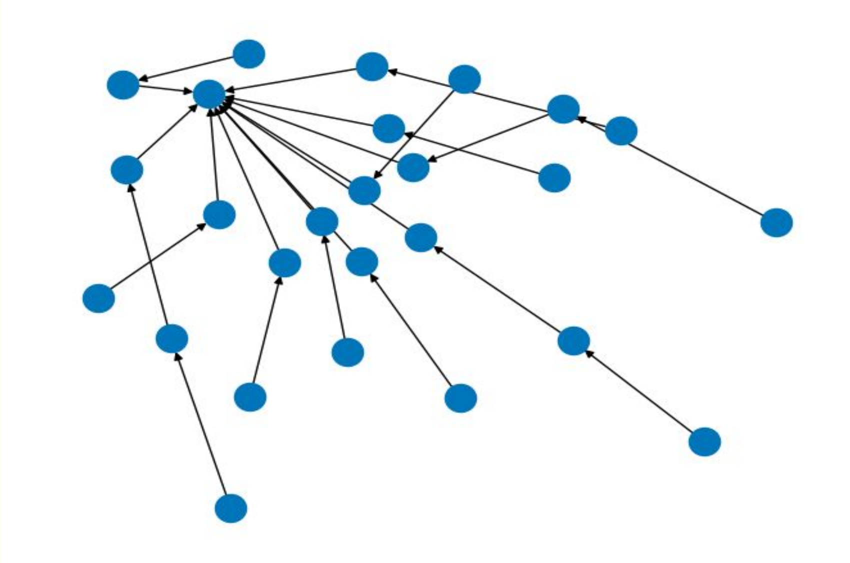
INTERPRETABILITY OF BERT MODEL



Conversation Graph

```
import networkx as nx
from torch_geometric.utils import to_networkx
G1 = to_networkx(dataset[119], to_undirected=False)
nx.draw(G1)
```

```
Number of training graphs: 198
Number of test graphs: 112
```



Graph Model: GCN

GCN is a type of convolutional neural network that can work directly on graphs and take advantage of their structural information. The general idea of GCN: For each node, we get the feature information from all its neighbors and of course, the feature of itself. Assume we use the aggregate() function. We will do the same for all the nodes. Finally, we feed these average values into a neural network.

Train Accuracy	**86.5**
Test Accuracy	**85.03**

GRAPH MODEL: GAT

GAT (Graph Attention Network), is a novel neural network architecture that operate on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations. Steps involved:

1. Simple linear transformation
2. Attention Coefficients
3. Softmax
4. Aggregation

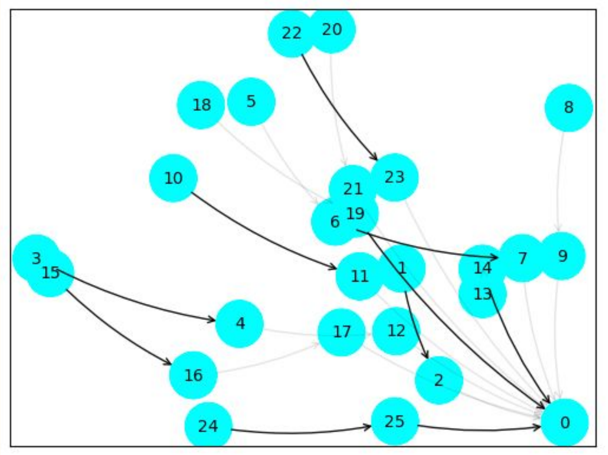
Train Accuracy	**87.00**
Test Accuracy	**86.38**

Graph Model Interpretability

Interpretability of the graph model is handled using **GNNExplainer.**

The black edges indicate that the following node provided more useful information towards that particular label in the classification of the graph.

The grey edges indicate that the information passed from the node to it's connected node was not very useful in classification.



Joint Model

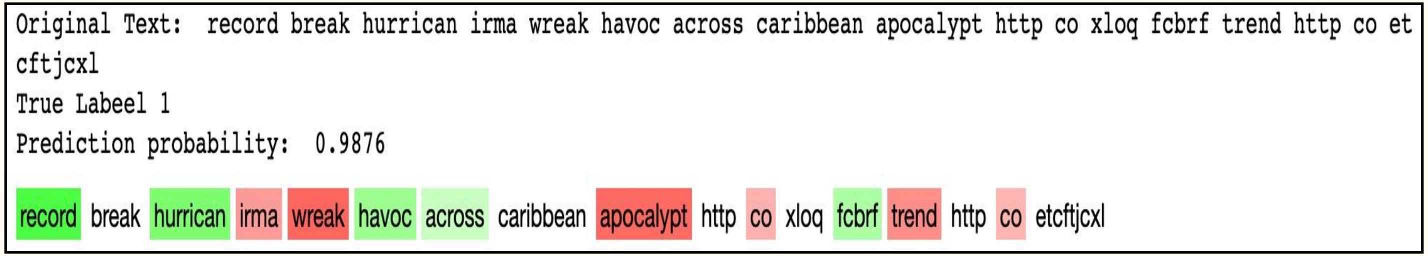
A joint model is one that combines two media types; in our case, it is text and graph. Our approach for creating the joint model consists of concatenating the embeddings from the text model, Bert and the graph model, GCN using PyTorch as shown below.

```
class Joint(torch.nn.Module):
    def __init__(self,bert,gnn):
        super(Joint,self).__init__()
        self.bert=bert
        self.gnn=gnn
        self.fc1=Linear(832,64)
        self.fc2=Linear(64,2)

    def forward(self,x1,x2):
        textemb=torch.tensor(plswork(x1))

        tokens_text = tokenizerIP.tokenize([str(x1).lower()])
        tokens_text, mask_text, _ = tokenizerIP.preprocess_classification_tokens(tokens_text, MAX_LEN)
        o1=interpretBERT.predict(token_ids=tokens_text,input_mask=mask_text, batch_size=64)
        o2,gemb=self.gnn(x2.x,x2.edge_index,batch=None)
        pred = o2.argmax(dim=1)  # Use the class with highest probability.


        x = torch.cat([textemb, torch.tensor(gemb[0])], dim=-1)
        x=self.fc1(F.relu(x))
        x=self.fc2(F.relu(x))
        return x,o1,pred
```

Results and Discussions

Text Model: EmbeddingBag Classifier

- The model was trained for 20 epochs and uses the ADAM optimizer with default parameters.
- This baseline model provided an accuracy of 80.5%.
- The interesting part of this model was the good explainability it provided. It almost accurately highlights informative words like 'hurricane', 'havoc',etc.

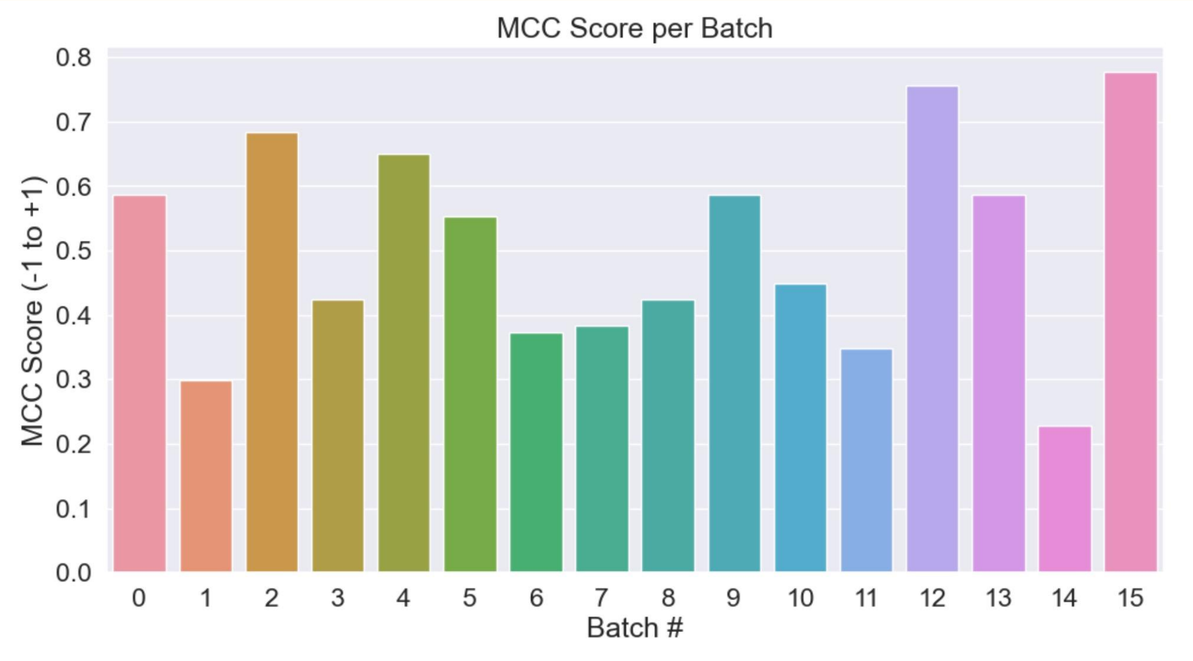


Results and Discussions

Text Model: BERT Sequence Classifier

- We train the model for 2 epochs and use the ADAM optimizer. It took approximately 5 hours to train the model for 2 epochs.
- We use Matthews Correlation Coefficient as a metric for the test set.
- We obtained a satisfactory MCC of 0.507 across batches.
- The explanation dashboard efficiently provided the influential top-k words for each prediction.

Results and Discussions



Total MCC	**0.507**

Text Model Results

- We see that BERT Sequence classifier performs much better than the simple embedding bag model.
- BERT uses a pre-training and fine-tuning approach, which allows it to be trained on a large corpus of text data and then fine-tuned for a specific task. This allows it to learn general-purpose language representations that can be applied to a wide range of tasks, which can improve its performance compared to a model like Embedding Bag that is only trained on a specific task.

Model	**Accuracy**
EmbeddingBag Classifier	80.5
Bert Sequence Classifier	85.00
Bert-MCC Score	0.507

Results and Discussions

Graph Model: Graph Convolutional Network

- The model has 2 graph convolutional layers and a linear layer for classification.
- It is trained for 50 epochs and uses ADAM optimizer.
- It gave a test accuracy of 85.03%

Graph Model: Graph Attention Network

- The model has 2 graph attention convolutional layers.
- It is trained for 50 epochs and uses ADAM optimizer.
- It gave a test accuracy of 86.38%

Joint Model Results

- We create a custom Pytorch module which combines the embeddings obtained from the text and graph to jointly learn from both views using 2 fully connected layers.
- We obtained an accuracy of 92.00% on the test set of tweets that have at least one or more replies to it.
- We observe that the joint model performs better than the graph and textual model when trained separately.

Discussion

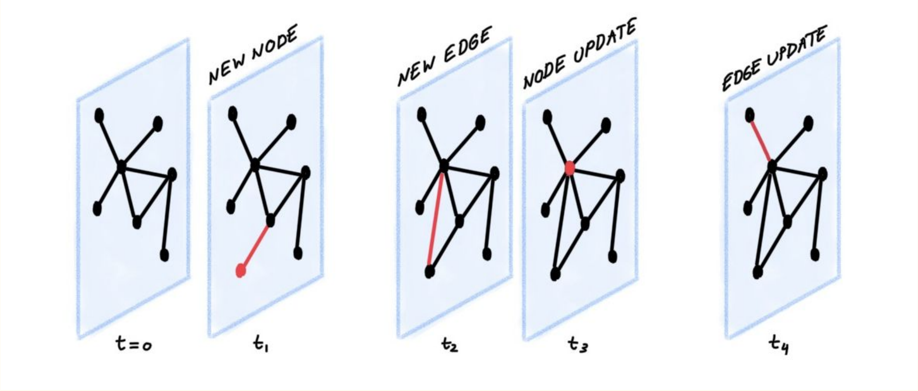
- We observe that joint learning can be effective for learning from multimodal data because it allows the model to learn a shared representation that captures the relationships between different modalities, improve its ability to generalize to new data, and be more efficient than other methods.
- We notice that GCN works better than GAT in the Joint Model.
- Our results support our initial hypothesis that adding conversation structure as a feature for classifying the tweet can provide more context and hence improve the accuracy.
- Interpretability of both, graph and text , helps us in understanding our models predictions.

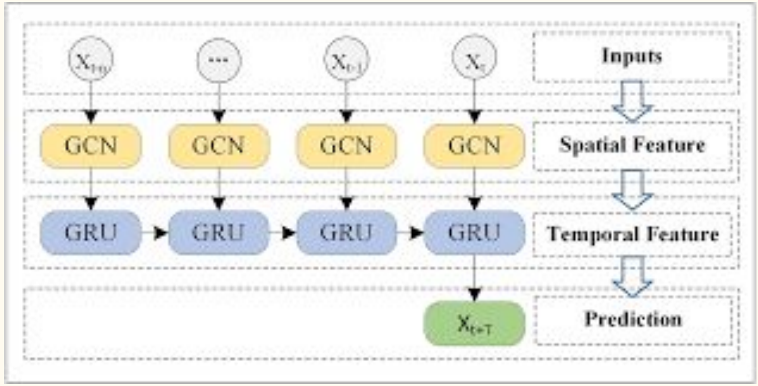
Conclusion

- A joint learning approach give higher accuracy than each of the models separately. Extracting information from both views and learning from both of them helps the neural network in better generalizing the task.
- Conversation graph enriches the purely NLP task by providing more context for classification, hence resulting in a more robust model.
- Interpretability of both the individual models' prediction helps us understand the reasoning behind the prediction.
- Twitter can be used as a source of valuable information for routing help to the people who need it during a disaster and our model can help in filtering out the the valuable tweets from the vast amount of data being generated during such times.

Future Work

- We aim to incorporate a temporal factor by creating snapshots of conversation graph in intervals to detect the level of emergency of the disaster, thus adding more functionality to the project other than the original two.
- We plan to extract embeddings of the individual snapshot and pass it to a RNN like LSTM/ GRU to learn the temporal dependencies in this graph as shown in the proposed architecture





References

[1] F. Alam, F. Ofli, M. Imran, και M. Aupetit, 'A twitter tale of three hurricanes: Harvey, irma, and maria', arXiv preprint arXiv:1805. 05144, 2018.

[2]W. Cui, J. Du, D. Wang, F. Kou, και Z. Xue, 'MVGAN: Multi-View Graph Attention Network for Social Event Detection', τ. 12, τχ. 3, 2021.

[3] C. Fan, F. Wu, και A. Mostafavi, 'A Hybrid Machine Learning Pipeline for Automated Mapping of Events and Locations From Social Media in Disasters', IEEE Access, τ. 8, σσ. 10478–10490, 2020.

[4] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, και W. Lu, 'STGAT: Spatial-Temporal Graph Attention Networks for Traffic Flow Forecasting', IEEE Access, τ. 8, σσ. 134363–134372, 2020.

References

[5] Y.-J. Lu και C.-T. Li, ‘GCAN: Graph-aware co-attention networks for explainable fake news detection on social media’, arXiv preprint arXiv:2004. 11648, 2020.

[6] H. Nam, J.-W. Ha, και J. Kim, ‘Dual attention networks for multimodal reasoning and matching’, στο Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, σσ. 299–307.

[7] R. Ying, D. Bourgeois, J. You, M. Zitnik, και J. Leskovec, ‘GNNExplainer: Generating Explanations for Graph Neural Networks’, Advances in neural information processing systems, τ. 32, σσ. 9240–9251, 12 2019.