

Text Classification, Interpretability, Robustness

Team Name: Okapi BM 2k22

Team Member: Debashish Roy (2021201034), Prudhvi (20173049), Sumeet Agarwal (2018900090)

Professor: Dr. Vasudeva Varma

Mentor: Mr. Vijayasaradhi

18th November 2022

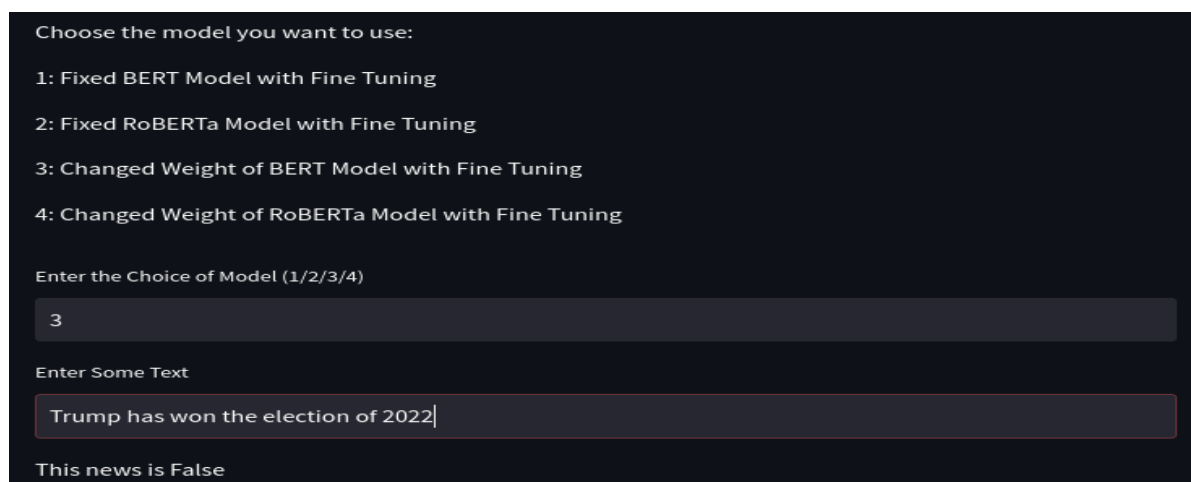
Link to the work: <https://github.com/debashish05/Fake-News-Detection/>

1. Introduction

Fake news detection is the task of identifying the intentional spread of misinformation through any form of broadcast. The aim of fake news is to cause disharmony in the community or make money from the news. The main problem with fake news is that they tend to be negative and as we all know fake news travel faster than good news and causes a significant threat to peace and stability.

Fake news is not a new concept. But with the rise of social media and increasing connectivity of the internet, communication speed has tremendously increased. So, the spread of false negative news speed also increased. There are fact checkers to check the authenticity of the news, the content on the internet is so vast, that it is impossible to monitor them manually. So we need an automated solution.

In this project, we are using pretrained language models and fine tuning them to predict the news is fake or not. News contents contain the clues to differentiate fake and trusted news. However, these works fail to consider different sentence interaction patterns between trusted and fake news documents, because only depending on the structure of the text is not always enough. So we developed a graph neural network accompanied by a knowledge graph for the prediction of news. Instead of only relying upon only the text in the news, we are using external knowledge as well. The text is compared with the external source as well to get more accurate results.



The screenshot shows a Streamlit web application interface with a dark theme. At the top, it says "Choose the model you want to use:". Below this, there are four numbered options: "1: Fixed BERT Model with Fine Tuning", "2: Fixed RoBERTa Model with Fine Tuning", "3: Changed Weight of BERT Model with Fine Tuning", and "4: Changed Weight of RoBERTa Model with Fine Tuning". Below the options, there is a text input field labeled "Enter the Choice of Model (1/2/3/4)" with the number "3" entered. Below that, there is another text input field labeled "Enter Some Text" with the text "Trump has won the election of 2022" entered. At the bottom, it displays the result: "This news is False".

Deployment of our model using streamlit.

2. Related Work

Generally, fake news detection usually focuses on news events while fact-checking is broader (Oshikawa et al., 2020). The approaches for fake news detection can be divided into two categories: social-based and content-based.

In earlier works, convolutional neural networks (CNN) and LSTM methods are combined for various text-based features. RNN and CNN-based methods to build propagation paths for detecting fake news at the early stage of its propagation.

One of the other works is based on FEVEROUS task. It's a shared task hosted on fever.ai and churned out many popular approaches to detect fake news given a claim. The claims were generated by altering Wikipedia sentences. The claims are classified as SUPPORTED, REFUTED or NOT ENOUGH INFO by annotators achieving 0.6841 in Fleiss.

Most of the real datasets are small in nature and hence, the need for bigger datasets has inspired the creation of FEVER dataset. Most of the datasets conducted in real didn't consider evidence or justification. Wang 2017 developed 12.8K claims in it's PolitiFact dataset but didn't contain any evidence.

Feverous dataset was presented in 2021. The main improvement was taking into data coming from tables, info boxes etc in Wikipedia page. Also this dataset has improved number of claims. In both FEVER and FEVEROUS dataset, entity matching and TFIDF approach was used to retrieve the evidence from the dataset.

Our work is to explore, fine tuning the existing models. Along with that we have improvised and use of State of the art models in CompareNet which was proposed by BUPT. Existing fake news detection model CompareNet, which directly compares the news to external knowledge for fake news detection. It considers topics for enriching news representation since fake news detection is highly correlated with topics.

3. Baseline Methodologies

3.1 Fine Tuning and Hybrid CNN

For the baseline in LIAR dataset we have used the best results mentioned in the research paper "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. They have use Hybrid CNN to get an accuracy of 0.274 by using Text+ and using All features. LIAR is a consists of 6 classes. This dataset contains a variety of fine-grained articles into six categories: false(0), half-true (1), mostly true(2), true(3), barely true(4) and pants-fire(5). This leads to a 6 way mutli class classification problem.

The baseline chosen for FEVER data is RoBERTa encoder with linear layer on the top. The model is trained with FEVER dataset and used to predict on the real claims' dataset.

For the baseline for our model that is using Knowledge based representation we have taken compareNet itself. And experimented on the architecture itself. Let's first discuss about the CompareNet architecture.

3.2 CompareNet

CompareNet considers topics for enriching news representation since fake news detection is highly correlated with topics. A directed heterogeneous document graph for each news document incorporating topics and entities is created. Let's take an example about how the directed heterogeneous graph is made. The graph consist of vertex topics, sentences and entities.

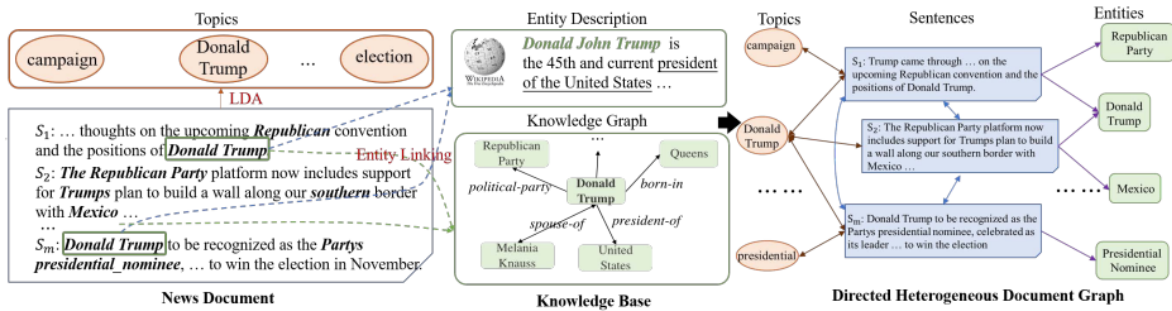
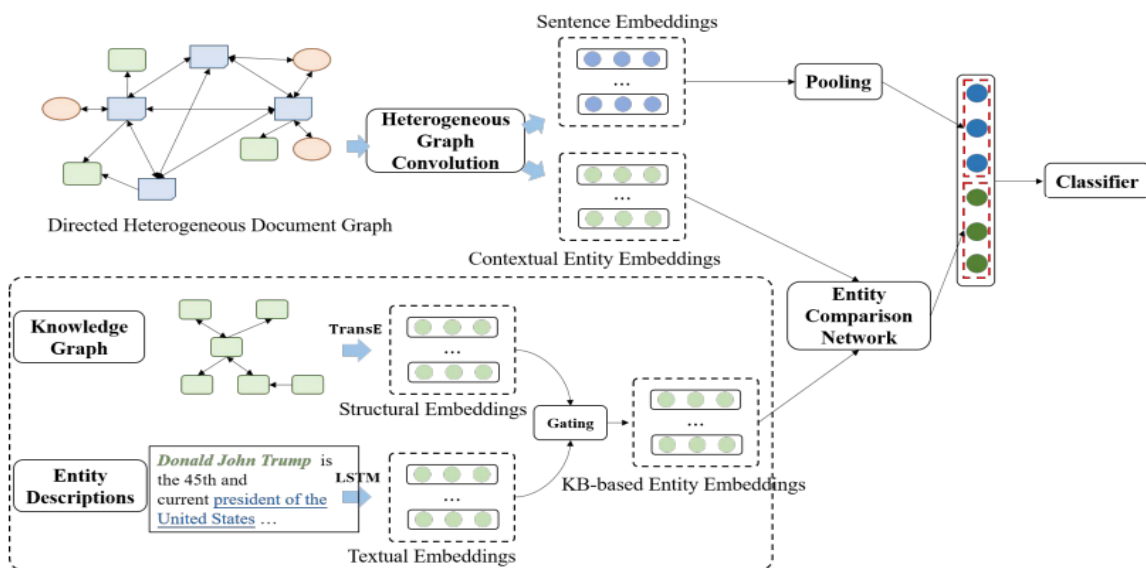


Figure 1: An example of directed heterogeneous document graph incorporating topics and entities.

The bidirectional edges are defined between topics and sentences. Unidirectional edges between entities and sentences. Bidirectional edges between sentences and sentences which are similar. Based on the graph, a heterogeneous graph attention network was built to learn the topic-enriched news representation as well as the contextual entity representations that encode the semantics of the news document.



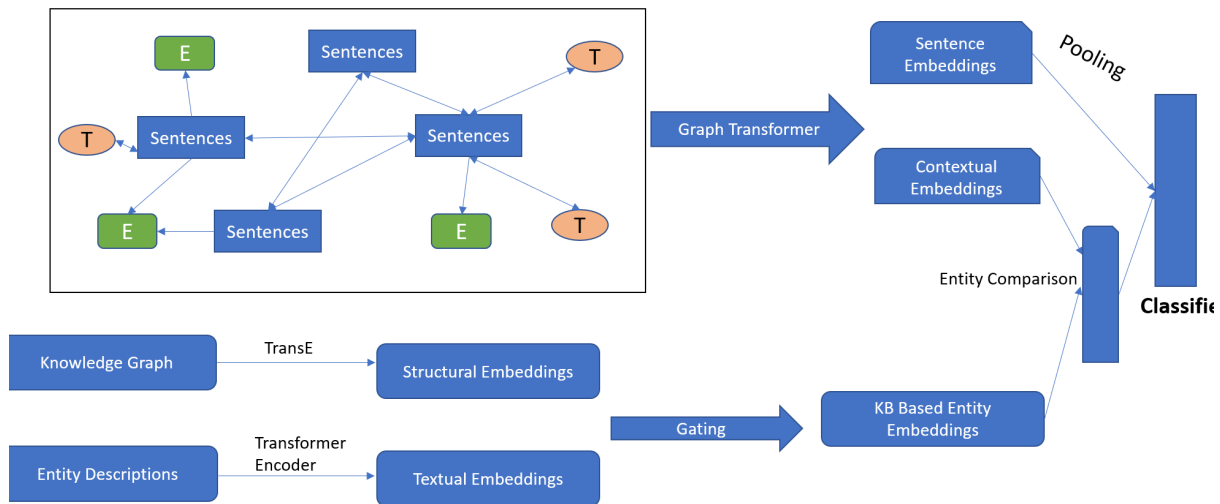
Baseline Architecture of compare NET

To fully leverage external KB, entities was taken as the bridge between the news document and the KB. Contextual entity representations was compared with the corresponding KB-based entity representations using a carefully designed entity comparison network. Finally, the obtained entity comparison features are combined with the topic-enriched news document representation for fake news detection.

4. Architecture

4.1 CompareNet++

We Explored the State of the Art models to improvise attention model.



Architecture of Modified CompareNet, i.e. ComapreNet++

We experimented Transformer Encoder Architecture with 6 layers instead of LSTM and used graph Transformer network instead of attention.

In Graph Transformer Layer we just used the encoder part and trained on the entire dataset. The parameters are trainable parameters. No positional Encodings are used for this experiment.

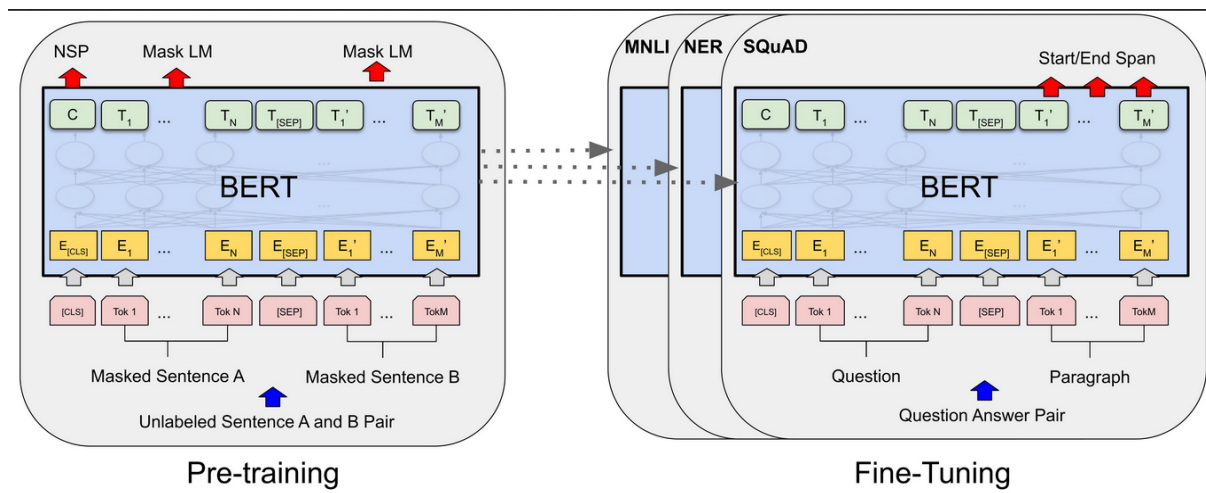


It is feasible to use to Graph TransformerNetwork where a node attends to local node similar to Graph Attention Networks which is being used in the existing paper.

Graph Transformer Networks (GTNs) have which preclude noisy connections and include useful connections (e.g., meta-paths) for tasks, while learning effective node representations on the new graphs in an end-to-end fashion.

4.2 Fine Tuning

We have used a BERT and RoBERTa based encoder model with linear layers to produce the output. We have experimented with fixing and changing the weights of the model.



Freeze language model: Freeze all the layers of the language model and attached neural network layers. The weights of only the attached layers will be updated during model training.

Train the entire architecture: Train the entire pre-trained model followed by neural network layer. In this case, the error is back-propagated through the entire architecture and the pre-trained weights of the model are updated.

5. Evaluation Mechanism & Results

We have experimented with Fine tuning the BERT and RoBERTa. At best it is beating the baseline model of Hybrid CNN, which is giving 0.274 accuracy by 0.29. The results of them on LIAR dataset are as follows:

	BERT			RoBERTa		
	Precision	Recall	F1- Score	Precision	Recall	F1- Score
False	0.30	0.16	0.21	0	0	0
Half-true	0.21	0.89	0.33	0.21	1	0.34
Mostly true	0	0	0	0	0	0
True	0	0	0	0	0	0
Barely true	0	0	0	0	0	0
Pants-fire	0	0	0	0	0	0
Accuracy			0.22			0.21
Macro Avg	0.09	0.17	0.09	0.03	0.17	0.06
Weighted Avg	0.10	0.22	0.11	0.04	0.21	0.07

RESULTS (FINE-TUNED WITH FREEZE BASE MODEL) on LIAR Dataset

	BERT			RoBERTa		
	Precision	Recall	F1- Score	Precision	Recall	F1- Score
False	0.26	0.41	0.32	0.29	0.40	0.34
Half-true	0.29	0.38	0.33	0.28	0.30	0.29
Mostly true	0.33	0.20	0.25	0.28	0.42	0.34
True	0.30	0.32	0.31	0.30	0.28	0.29
Barely true	0.27	0.10	0.14	0.25	0.05	0.08
Pants-fire	0.32	0.27	0.29	0.44	0.21	0.28
Accuracy			0.29			0.29
Macro Avg	0.29	0.28	0.27	0.31	0.28	0.27
Weighted Avg	0.29	0.29	0.27	0.29	0.29	0.27

RESULTS (FINE-TUNED WITH Changed Weight of BASE MODEL) on LIAR Dataset

Metric	Roberta on FEVER (paper results)	Roberta on FEVER (replicated on smaller dataset)
F1 – Supported	0.89	0.74

F1 - Refuted	0.87	0.71
F1 -Misleading	0.05	0.01

Roberta trained on FEVER dataset

Metric	Roberta on FEVER	Roberta predictions on fake claims before finetuning
F1 – Supported	0.74	0.59
F1 - Refuted	0.71	0.51
F1 -Misleading	0.01	0.02

Trained ROBERTa prediction results on Fever

Metric	Before finetuning, training on claims	After finetuning, retraining
F1 – Supported	0.59	0.67
F1 - Refuted	0.51	0.62
F1 -Misleading	0.02	0.02

ROBERTa Training results on real claims dataset

CompareNET++ Results:

The compareNet++ is tested on Labelled Unreliable News Dataset and the results are as follows:

Accuracy on the test set 2: 0.2516

	Precision	Recall	F1 Score
Macro Avg	0.0630	0.2490	0.1005
Micro Avg	0.2516	0.2516	0.2516

6. Analysis

6.1 CompareNET++ with Transformer Encoder and Graph Transformer Network layer

We have performed the experiment on 48000 Training records and 2k test records. The results show that adding GTN doesn't significantly improve the precision and recall and with lower learning rate and increasing epoch, overfit the training data.

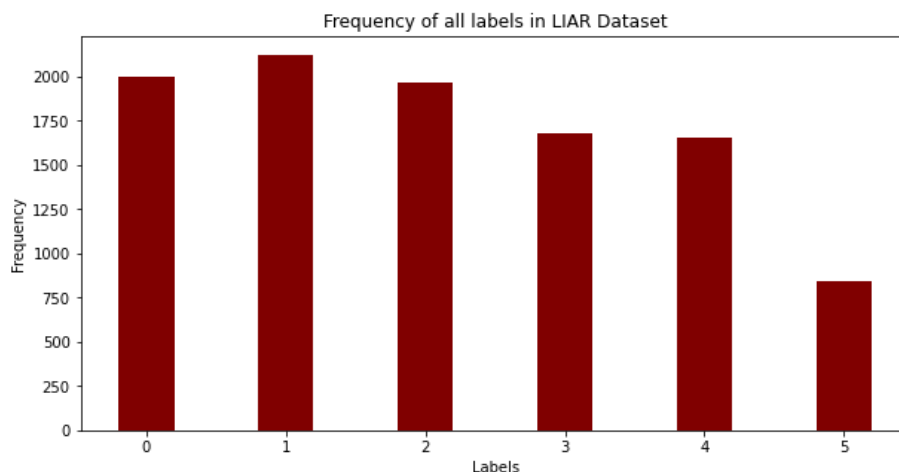
Adding a Transformer layer for LSTM is more relevant to avoid long term dependency problems. Improving the classifier with more layers and adding pretrained Bert model for text embedding is the future course of work we are working on to improve the fake news detection.

6.2 Fine Tuning

6.2.1 LIAR Dataset

We have finetuned two encoder based models BERT and RoBERTa. The analysis are as follows:

- **Fixing the weights of the Base Model:** When fixing the weights of the base model only the last few layer get a chance to adjust the weights. Interestingly these layer always tries to predict the label which is occurring more frequently. For example, these are the frequency of each label in the entire LIAR dataset.



Label no. corresponds to false(0), half-true (1), mostly true(2), true(3), barely true(4) and pants-fire(5).

So the model always BERT model always keep on telling the news label is either 0 or 1, on the other hand RoBERTa will keep on telling that the news is of label 1, irrespective of the content. Because this is straight forward way to achieve a good accuracy considering a 6 class classification problem.

- **Changing the weights of the Base Model:** Since this time we allowed the base model to weights, the model is giving a diverse set of outputs. And giving a better accuracy in general. This model outperformed the baseline mentioned in the LIAR paper and gave an accuracy of 0.29.

6.2.2 FEVER and Real Claims dataset

The ROBERTa based encoder model with a linear layer was used to predict the results. The results were not that promising owing to two reasons.

- Roberta model trained on Wikipedia data isn't well trained for numerical reasoning, multi hop reasoning. Most of the real-life data has a lot of the above.
- Fully trained model required a lot of space which we didn't have on current laptop where we trained. So it's our understanding that training with higher data can better the results.

Post the above result, we fine tuned the parameters with number of linear layers. We used half of the dataset to retrain the model and used it for prediction. As per the table, we observed improvement after finetuning and training on real dataset.

Dataset creation to train transformer models is the founding stone for FEVER dataset. But these large datasets however large they are, doesn't include examples on numeric reasoning, multi hope reasoning. The real life datasets of fake claims, verification include a lot of numeric, multi hop types.

We demonstrated that the models trained well on FEVER datasets were not able to perform well on real claims dataset. With finetuning and retraining, the model accuracy is improving. We believe that all models / approaches being developed on FEVER kind of datasets also need to evaluate on real claim datasets.

6.3 Claim, Evidence, Verdict Extraction for real claims dataset for FEVER

6.3.1 Research Questions and Data Collection

Many baselines and interesting approaches were built. But none of the trained models were used to test on real world datasets. We had following RQs

RQ1: *How is the performance of pretrained models on real world claims dataset.*

RQ2: *Can the performance of the pretrained models be improved by Fine tuning the real world claims dataset.*

To continue with our research question, we need data from real world in the format of claim, evidence and verdict. For this we have chosen the following website – FACTLY, Alt news, BoomLive, OpIndia. We built web scrapper to extract article title, content, web url. A total of 10,534 articles were extracted.

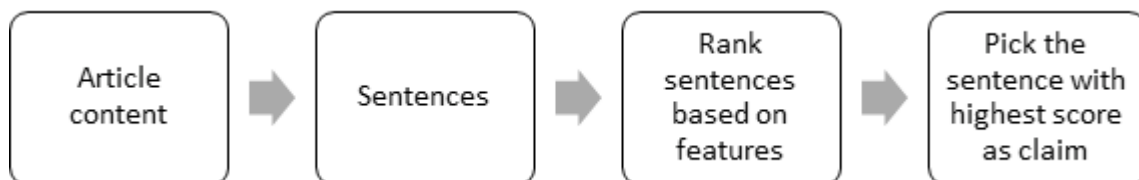
Sitename	Article Count
Factly	2421
Alt News	2957
Boomlive	3748
OpIndia	1408
Total	10,534

Given the complex nature of each of the claim, verdict, evidence, we used different approaches to extract each one of the following.

6.3.2 Claim Extraction

We observed that in general, the title of the article gave the most clues about claim. The article content is scrapped and tokenised into sentences. For each sentence, we computed a claim score – number of unigrams, bigrams matching with title, position (sentence index) in the article, presence of key words similar to ‘claim’, ‘claimed’. The sentence with highest score is used claim. In general, it’s based on observation that most of claims are produced as is in first paragraph, and has the words matching – ‘claim’ and most likely overlapping with title.

Evaluations – Once the output is generated, we manually verified it for about 50 articles and found them to accurate claims



6.3.3 Evidence Extraction

For evidence extraction, we compare each sentence with the claim. We compared with entity matching, TFIDF based similarity score, Unigram and Bigram matching. We pick up the sentences only from second or greater para to avoid picking other claims. WE extracted top3 sentences as evidence.

It is to be noted while this is not perfect way of picking up evidence, more tricks, techniques need to worked out to improve efficient evidence extraction



6.3.4 Verdict extraction

For verdict extraction, we used FACTLY data as a base. Most verdicts are present in a tagged format. We used html based scrapping and identified the pattern. The tags we obtained were – TRUE, FALSE, MISLEADING, UNVERIFIED. We searched for similar words in the content and tagged the claim as one of the above. A sample data looks like this

Title of article	Claim	Evidence	Verdict
The sword in this photo does not belong to Maharana Pratap	Image of sword used by Indian ruler Maharana Pratap.	The sword in the image does not belong to Indian ruler Maharana Pratap. It belongs to Boabdil (also known as Muhammad XII), the last Nasrid ruler of the Emirate of Granada. So, the claim made in the post is FALSE.	FALSE
X-ray showing a live cockroach in a patient's chest is a photoshopped one	In Zimbabwe, an X-ray image shows a live cockroach in a patient's chest.	The image in the post is a photoshopped one. The source image is an X-ray of Marilyn Monroe's chest taken by a doctor at Cedars Hospital of Lebanon in 1954. So, the claim made in the post is FALSE.	FALSE
Old photo of a road in Bihar is falsely shared as the condition of roads in Uttar Pradesh	Photo showing the condition of roads full of potholes in Mahul village, Uttar Pradesh.	Image in the post shows the old condition of Bhagalpur-Pirpainty-Mirzachowki National highway-80 at Bhagalpur district in Bihar. It is not from the Mahul village in Uttar Pradesh state. Hence the claim made in the post is FALSE.	FALSE.

In total, we were able to extract 9,875 articles. For training purpose, we used 2,580 articles by FACTLY which are considered to be gold standard.

7. Correction from the Feedback Recieved

- Roberta Model trained accurately on train data.
- Other solutions from FEVER dashboard explored - working codes not found. So I stuck to working with baseline with fine tuning approach.
- Converted the results into a more readable format for precision, recall and F1-score.

8. Reference

- <https://aclanthology.org/2021.acl-long.62.pdf> CompareNet.
- https://github.com/BUPT-GAMMA/CompareNet_FakeNewsDetection
- https://github.com/MysteryVaibhav/fake_news_semantics
- FEVEROUS: Fact Extraction and Verification Over Unstructured and Structured information by Rami Aly et.al arXiv:2106.05707 <https://github.com/Raldir/FEVEROUS>
- Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking](<https://aclanthology.org/D17-1317>) (Rashkin et al., EMNLP 2017)
- Liar, Liar Pants on Fire <https://arxiv.org/pdf/1705.00648v1.pdf>
- https://github.com/pyg-team/pytorch_geometric Pytorch Geometric
- <https://link.springer.com/article/10.1007/s40747-021-00552-1>
- <https://aclanthology.org/D17-1317/>
- <https://paperswithcode.com/task/fake-news-detection>
- https://www.youtube.com/watch?v=LbYF0yMIFaM&list=PL83F70cPvROZeFvtp7XWx8tvql_eJpiop&index=5
- <https://huggingface.co/datasets/liar/viewer/default/train> LIAR Dataset
- [Graph Transformer Networks \(neurips.cc\)](https://neurips.cc/paper/2017/graph-transformer-networks)
- [GitHub - graphdeeplearning/graphtransformer: Graph Transformer Architecture. Source code for "A Generalization of Transformer Networks to Graphs". DLG-AAAI'21.](https://github.com/graphdeeplearning/graphtransformer)