Resolving the Spread of Fake News by Utilizing Natural Language Processing Tools to Fact-Check Information

Project Group 23
(Sam Edwards, Ross Tebbetts, Naseela Pervez, Daniel Bai, Hang Yang)

Motivation and Problem Definition

- Motivation
  - Spread of misinformation has increased with the advance of technology.
  - Social media like Twitter, Facebook etc have been the hub of spread of rumours.
  - Combating the spread of false information can help mitigate the spread of fake news and rumours as well as cybercrimes.
  - Companies such as Twitter and Instagram, will often now add a caution note over the image, warning app users that the information may or may not be credible.

- Problem Statement
  - We felt that developing a functional Fact-Checking model would be a great application of NLP from this course, and application to the integrity issues faced in today's society.
  - The dataset that we have used for the same is FEVER-19 dataset which classifies claims as SUPPORTS, REFUTES and NOT ENOUGH INFO based on the evidences provided.
  - The problem when broken down to the simplest form is a multiclass classification problem.

Proposed Solution

Preprocessing steps:

- Extracting evidence texts from the wikipedia corpus.
- Keeping only English language words in claims and evidences

Keeping the resources and the extent of the project in mind, we carried out the following experiments:

- Classical Machine Learning Model (Linear SVM, Multinomial Naive Bayes) with TF-IDF features of claims and evidence_text
- Perceptron Model with TF-IDF features of claims and evidence_text.
- RNN with TF-IDF features of claims and evidence_text.
- Bi-LSTM Model using learnt GloVe Embeddings
- BERT Model

The evaluation metrics that we focused of were - Accuracy, F1 score and Recall Score.

We used the traditional 80-20 split for the evaluation of the classical machine learning models, perceptron and RNN.

For Bi-LSTM and BERT we evaluated the models on the dev data provided by FEVER.

Results

Comparing the classical machine learning models Naive Bayes and SVM, we can clearly see that the F1 Score of SVM is much higher than Naive bayes as well as the other evaluation metrics.

In the earlier stages of the course, it was stated that SVM performs better with text data and the same results can be verified here. Please note that TF-IDF has been used for feature extraction

We also experimented with a single layer perceptron model. The performance of the model did not surpass the classical SVM model.

Among the deep learning models, we experimented with BiLSTM model and Pretrained BERT model using Glove Embedding and BERT embeddings respectively. For the BiLSTM model, the F1 score on dev data is 0.88 and the recall is 0.89 which is much higher than the rest of the models.
Due to limitation of the computational resources we fine tuned BERT on a smaller subset of the dataset.

For model parameters, we experimented with many parameters and chose the ones which gave the best results. The same has been followed for the training parameters.

In conclusion, the most accurate model was the BiLSTM Model, yielding the highest F1, Accuracy, and Precision Scores. The team took the project a step further, to develop a User Interface tool which can be used to prompt the model with a claim, and assess the accuracy of that claim.

Model Results:

	Accuracy	F1-Score	Recall Score
Naive Bayes	0.65	0.58	0.65
SVM	0.84	0.82	0.84
Perceptron	0.80	0.80	0.80
RNN	0.55	0.71	0.69
BiLSTM	0.89	0.88	0.89
BERT	0.67	0.55	0.67

Model Parameters:

Parameters	BiLSTM
Num Layers	1
Dropout	0.4
Embedding Dimension	100
Hidden Dimension	256
Output Dimension	128

Training Parameters:

Parameters	BiLSTM
Epochs	10
Learning Rate	0.01
Loss Function	CrossEntropyLoss
LR Scheduler	ReduceLROnPlateau
Optimizer	Adam