Efficient detection of fake news stances using Deep Learning Techniques

Problem Statement:

Fake News or deception is an emerging topic that has received a lot of attention since the 2016 US presidential election, where many reckon that the spread of false information on social networks had a significant influence on its outcome. Most of the current work focuses either on the actual content of the news articles or the user that shares the news article on social media. However, social media platforms where fake news spread can be easily modelled as graphs and the goal of this project is to leverage techniques from deep Learning on Graphs for design better models for fake news detection.

Challenge: - Develop an efficient deep neural network for detection of deceptive content using graph properties

Early studies attempted to detect fake news based on linguistic features extracted from news articles with traditional machine learning models. Some Recent studies have used deep learning models to capture only the content and context side using recurrent neural networks (RNNs) and convolutional neural networks (CNNs). Some researchers have used recursive neural networks based on the texts content only with different writing styles. In a Graph-based dataset, graph-based connections for a news articles with the other features are also useful for fake news detection. We have to design a model to handle/utilizes linguistic, user, and more context features connected a graph like structure in the dataset.

Experimental Design

Dataset: - Twitter15 and Twitter16

The datasets we work with are Twitter15 and Twitter16. These two datasets share the same exact structure. Both of them contain the tweets and re-tweets from a thousand of news articles published in 2015 and 2016. For each news article, the data contains the first tweet that shared it on Twitter, and a sequence of re-tweets following this initial post. Each event is labeled according to the initial news article, the label is taken out of four possible classes: "true", "false", "unverified", "non-rumor". Labels are evenly distributed in both datasets.

Link: -

https://www.dropbox.com/s/7ewzdrbelpmrnxu/rumdetect2017.zip?dl=0&file_subpath=%2Frumor_detection_acl2017

Evaluation Measures

Evaluation is measured in terms of Precision, Recall, F1-Score, Accuracy, false positive cases, and fake negative cases.

Software and Hardware Requirements

Python-based Deep Learning libraries will be exploited for development and experimentation of the project. Tools such as Anaconda Python, and libraries such as, Tensorflow, and Keras will be utilized for this process.

Previous benchmark Results: -

In previous benchmark results, there is a large scope to improve the results using deep learning models with Twitter15 and Twitter16.

	Twitter15			**Twitter16**		
Split	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
Recursive Tree[8]	NA	NA	0.723	NA	NA	0.737
RNN+CNN[3]*	NA	NA	0.842	NA	NA	0.863
GBDT_user	0.962	0.629	0.628	1.00	0.671	0.647
GBDT_seiz	0.672	0.412	0.360	0.741	0.506	0.377
Ens_GBDT	0.959	0.635	0.577	0.995	0.617	0.618
MLP text	0.931	0.568	0.536	0.882	0.634	0.549
LSTM text	0.899	0.584	0.622	0.922	0.622	0.587
GraphSage text	0.954	0.624	0.622	0.866	0.756	0.712
GCN all (Our best)	1.00	0.719	0.690	0.859	0.841	0.750

Mentor Name: Rohit Kumar Kaliyar