Fake News Classification

Input Data: Statement + Justification.

Output: Label.

Different Classification Model Tried:

1. Logistic Regression
2. Naïve Bayes
3. SVM
4. SGDClassifier
5. Random Forest
6. GradientBoosting
7. Biderectional LSTM with Conv layers.

Different Ideas:

1. Did some exploratory analysis to check missing values and class balance.
2. Started with vanilla classifiers on the statement from speakers.
3. Switched to using both text and justification for the classification.
4. After trying vanilla classifiers, I started building a Bi-directional LSTM model as talked in the paper. Used Glove vectors as embeddings.
5. Also tried using pretrained BERT and XLNET with spacy/Pytorch transformers to tokenise and encode the text into word vectors. It did not give good result and was very compute heavy.
6. Experimented with using speaker name, job, state and other category-based features into the model. Just added their values to the text columns and allowed embedding layers to learn their importance. Needed more compute to effectively train embeddings for this task.

Highest-Accuracies:

Binary Classification: 60%

6-Way Classification: 22% [Naïve bayes , and BI-directional LSTM]

Instructions To Run.

Easy-Way with smaller compute and small batch size. [Preferred Way]

1. Upload the jupyter notebook to the Google Collaboratory.
2. Set the runtime as GPU. [It will allocate K-80 GPU]
3. Run the code in sequence.

Hard-Way with better compute and bigger batch size.

1. Create a account on Google clound.
2. Check out this for refence https://course.fast.ai/start_gcp.html
3. Use the following commands to setup vm.

```
export IMAGE_FAMILY="pytorch-latest-cpu"
export ZONE="us-west1-b" # budget: "us-west1-b"
export INSTANCE_NAME="hck-fast"
export INSTANCE_TYPE="n1-highmem-8" # budget: "n1-highmem-4"


gcloud compute instances create $INSTANCE_NAME \
        --zone=$ZONE \
        --image-family=$IMAGE_FAMILY \
        --image-project=deeplearning-platform-release \
        --maintenance-policy=TERMINATE \
        --accelerator="type=nvidia-tesla-v100,count=1" \
        --machine-type=$INSTANCE_TYPE \
        --boot-disk-size=200GB \
        --metadata="install-nvidia-driver=True" \
        --preemptible


gcloud compute ssh --zone=$ZONE jupyter@$INSTANCE_NAME -- -L 8080:localhost:8080
```

4. Add anaconda to the path. Install libraries specified in https://github.com/rishabmps/Fake-News-classification/requirements.txt.

5. Other documents will also be uploaded on https://github.com/rishabmps/Fake-News-classification

6. Upload the notebook.

7. Run in sequence.

Citations:

[1]. https://github.com/nishitpatel01/Fake_News_Detection For vanilla classification.

[2]. https://github.com/bedarkarpriyanka/NLP-Project-Fake-News-Detection Custom Embedding Layer.

[3]. https://keras.io/ Code samples.