All are not liars: Fine-grained fact-checking in political statements

Members: Yixuan Meng, Haoran Liu, Andrew Song

Penn Engineering

Contents

Motivation

- Low cost, easy access, and rapid dissemination of information lead people to seek out and consume news from web sources, but these exact sources enable the widespread of low quality news with intentionally false information.

- The extensive spread of fake news has the potential for extremely negative impacts on individuals and society

Difficulties with the Task

- Fake news detection on social media presents unique characteristics and challenges that make existing detection algorithms from traditional news media ineffective or not applicable

- Fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content

- Dynamic information such as how fake news and true news propagate and how users' opinions toward news pieces are very important for extracting useful patterns for fake news detection and intervention

How does this problem relate to CIS 530?

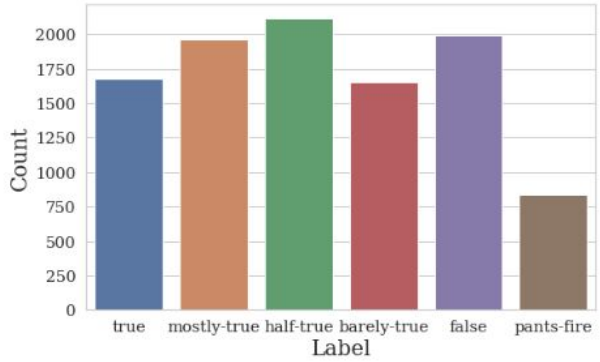
- Fake news detection is a critical NLP task that has gained rapid popularity with recent political events.
- Nontrivial labels and feature extraction make this an interesting and difficult problem that concepts and algorithms learned in CIS 530 will help tackle.
- Modern technologies are at the forefront of this field (ie neural networks).

Data

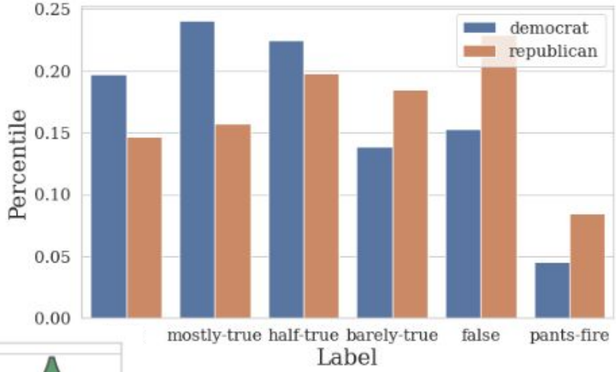
- The dataset of interest is the LIAR dataset, which includes 12.8 thousand rows of **short statements** that are labeled into six categories: pants-fire, false, barely-true, half-true, mostly-true, and true, and includes **metadata** of the text.

Summary Statistics	
Training Set Size	10,269
Validation Set Size	1,284
Test Set Size	1,283
Avg. Statement Length	17.9
Top 3 Speaker Affiliations	
Democrats	4,150
Republicans	5,687
None (eg FB posts)	2,185
Distinct Counts (Training)	
Speakers	2911
Contexts	4346
Subjects	3828

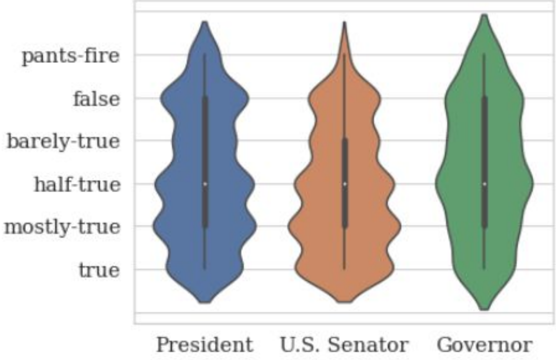
Data



2000
1750
1500
1250
1000
750
500
250
0
Count
true
mostly-true
half-true
barely-true
false
pants-fire
Label

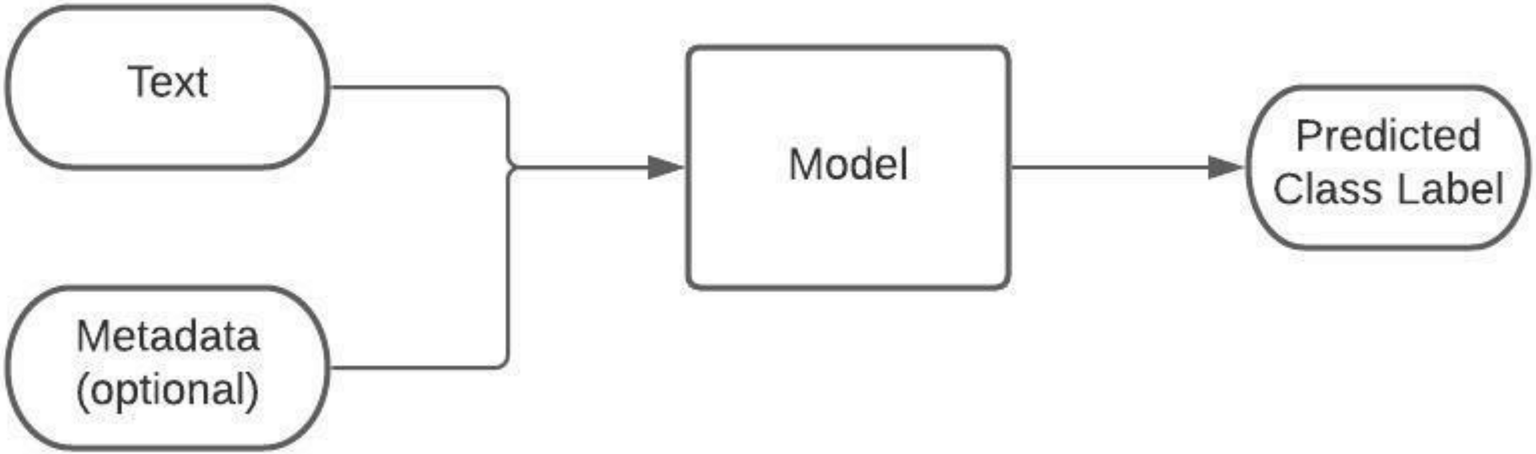


0.25
0.20
0.15
0.10
0.05
0.00
Percentile
democrat
republican
mostly-true
half-true
barely-true
false
pants-fire
Label



pants-fire
false
barely-true
half-true
mostly-true
true
President
U.S. Senator
Governor

Goal



Prior Models Used for Fake News Detection

- A plethora of models have been used, mainly on text-based methods not incorporating metadata.

Models	Valid.	Test
Majority	0.204	0.208
SVMs	0.258	0.255
Logistic Regress0ion	0.257	0.247
Bi-LSTMs	0.223	0.233
CNNs	0.260	0.270

(6 way classification for popular algorithms w/o metadata)

Evaluation Metrics

- Main evaluation metric cited in papers is **accuracy.**
- Many other papers support the use of **Macro F1 score.**

$$\text{Macro F1-score} = \frac{1}{N}\sum_{i=0}^{N} \text{F1-score}_i$$

Penn Engineering

Simple Baseline Model

- Majority classifier naively predicts the class that was seen most often in the training set

- The predictions will all be the same class.

- Accuracy ~ 19.3% on test set by predicting all as half-true.

Strong/Published Baseline Model

- CNN Model using word2vec embeddings with 300 dimensions
- Three convolution layers in the model, filter sizes 5,5,5 separately. Each size has 128 filters.
- The dropout rate is 0.8, and we concatenate the output of the three convolution layers and feed it to fully connected layer to do the six-way classification.
- For training, we use the Adam optimizer, set the learning rate to 0.001, use cross entropy loss and train for 10 epochs. The accuracy of this CNN model is **0.2368** on the validation dataset, and **0.2182** on the test dataset.

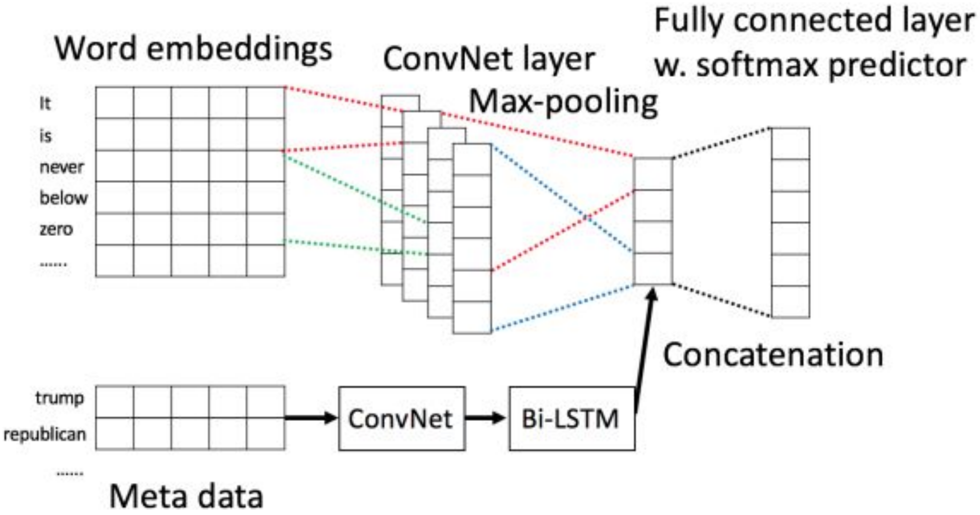
Extension Model #1

- We change the model structure and implement a BiLSTM with attention model. We also add a new metadata column `justification' from the Liar-Plus dataset, along with metadata from original dataset.

- We use the GloVe embedding for the metadata, and the Word2Vec embedding for the statement text, and train the BiLSTM with attention model with batch size 64, learning rate 0.0002 for 10 epochs, we get the best performance on the test set of 27.62% accuracy,

- We apply late fusion of metadata and statement embeddings

Discussion of Early vs Late Concatenation



Extension Model #2

- Our second extension incorporates the XLNet-base-cased pretrained model from HuggingFace.

- We use XLNetTokenizer and XLNetModel from transformers package, AdamW optimizer, and cosine scheduler using warmup.

- We add a fully connected layer and only update the parameters of the last four layers and freeze all other layers. Our model had a learning rate of .00015, batch size of 16, and ran for 15 epochs for an accuracy of **.2794**

Conclusions

- Achieved state-of-the-art accuracies on both extension models, .276 on extension #1 and .2794 on extension #2.

- Weighed intuition on the benefits of using early concatenation of data

- Interesting moral concern: Does using metadata discriminate certain groups?