Fake News Detection in Politifact texts using WorldNews Glove Vectors and MultiClass Logistic Regression

This task contains three sections: the state-of-the-art result for the dataset; the model that achieve those results; how I would improve on the results.

There are five papers will be referenced for state-of-the-art results in this task:

1. Guo, Z., Schlichtkrull, M., & Vlachos, A. (2022). A survey on automated fact-checking. Transactions of the Association for Computational Linguistics, 10, 178-206.
2. Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is Your Evidence: Improving Fact-checking by Justification Modeling. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 85–90, Brussels, Belgium. Association for Computational Linguistics.
3. Hossain, Md. Muzakker & Awosaf, Zahin & Prottoy, Md. Salman & Alvy, Abu & Morol, Md. Kishor. (2022). Approaches for Improving the Performance of Fake News Detection in Bangla: Imbalance Handling and Model Stacking.
4. Upadhayay, B., & Behzadan, V. (2020, November). Sentimental LIAR: Extended Corpus and Deep Learning Models for Fake Claim Classification. In 2020 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 1-6). IEEE.
5. H. E. Wynne and K. T. Swe, "Fake News Detection in Social Media using Two-Layers Ensemble Model," 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2022, pp. 411-414, doi: 10.1109/ITC-CSCC55581.2022.9894967.

Background: This dataset is collected from fact-checking website PolitiFact through its API. It includes 12,836 human-labeled short statements, which are sampled from various contexts, such as news releases, TV or radio interviews, campaign speeches, etc. The labels for news truthfulness are fine-grained multiple classes: pants-fire, false, barely-true, half-true, mostly true, and true.

Existing data model and explanation:

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
Hybrid CNNs		
Text + Subject	0.263	0.235
Text + Speaker	**0.277**	0.248
Text + Job	0.270	0.258
Text + State	0.246	0.256
Text + Party	0.259	0.248
Text + Context	0.251	0.243
Text + History	0.246	0.241
Text + All	0.247	**0.274**

The paper introduces five baselines: a majority baseline, a regularized logistic regression classifier (LR), a support vector machine classifier (SVM), a bi-directional long short-term memory networks model (Bi-LSTMs), and a convolutional neural network model (CNNs).

SotA and Model Explanation:

Muzakker Hossain [3] introduce three feature extraction for improving the result: COUNT VECTORIZER, TF-IDF VECTORIZER, WORD EMBEDDING.

TABLE I. CLASSIFICATION METRICS WITH COUNT VECTORIZER

Classifiers	Performance evaluation			
	Accuracy	*Precision*	*Recall*	*Fscore*
SVM Linear	0.91	0.90	0.91	0.90
Logistic Regression	0.93	0.91	0.92	0.92
Decision Tree	0.82	0.80	0.81	0.80
Random Forest	0.88	0.94	0.79	0.86
XG-Boost	0.89	0.88	0.88	0.88
Gradient Boosting	0.89	0.89	0.88	0.88
Neural Network	0.94	0.94	0.93	0.93

TABLE II. CLASSIFICATION METRICS WITH TF-IDF VECTORIZER

Classifiers	Performance evaluation			
	Accuracy	*Precision*	*Recall*	*Fscore*
SVM Linear	0.94	0.93	0.93	0.93
Logistic Regression	0.93	0.93	0.91	0.92
Decision Tree	0.82	0.79	0.80	0.80
Random Forest	0.90	0.94	0.83	0.88
XG-Boost	0.89	0.89	0.88	0.88
Gradient Boosting	0.90	0.89	0.88	0.88
Neural Network	0.93	0.93	0.91	0.92

TABLE III. CLASSIFICATION METRICS WITH WORD EMBEDDING

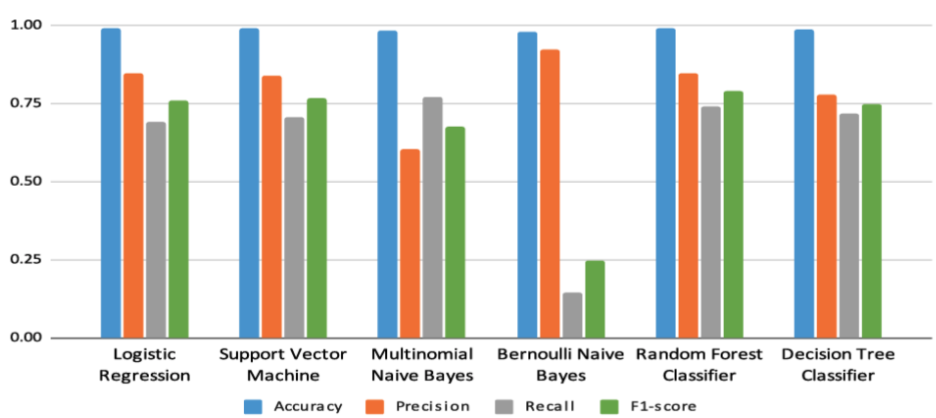
Classifiers	Performance evaluation			
	Accuracy	*Precision*	*Recall*	*Fscore*
SVM Linear	0.87	0.88	0.83	0.85
Logistic Regression	0.87	0.88	0.82	0.85
Decision Tree	0.73	0.64	0.69	0.69
Random Forest	0.84	0.85	0.79	0.82
XG-Boost	0.83	0.84	0.76	0.80
Gradient Boosting	0.83	0.84	0.76	0.80
Neural Network	0.90	0.92	0.86	0.89

Tariq Alhindi[2] extend the LIAR dataset by automatically extracting the justification from the fact-checking article used by humans to label a given claim.

Cond.	Model	Binary		Six-way	
		valid	test	valid	test
S	LR	0.58	0.61	0.23	0.25
	SVM	0.56	0.59	0.25	0.23
	BiLSTM	0.59	0.60	0.26	0.23
SJ	LR	0.68	0.67	0.37	0.37
	SVM	0.65	0.66	0.34	0.34
	BiLSTM	0.70	0.68	0.34	0.31
	P-BiLSTM	0.69	0.67	0.36	0.35
S^+M	LR	0.61	0.61	0.26	0.25
	SVM	0.57	0.60	0.26	0.25
	BiLSTM	0.62	0.62	0.27	0.25
S^+MJ	LR	0.69	0.67	0.38	0.37
	SVM	0.66	0.66	0.35	0.35
	BiLSTM	0.71	0.68	0.34	0.32
	P-BiLSTM	0.70	0.70	0.37	0.36

Table 2: Classification Results

While originally intended as an auxiliary task to improve claim verification, these justifications have been used as explanations (Atanasova et al., 2020b). Recently, Kotonya and Toni (2020b) constructed the first dataset which explicitly includes gold explanations. These consist of fact-checking articles and other news items, which can be used to train natural language generation models to provide posthoc justifications for the verdicts.



Bibek Upadhayay [4] introduce two model to improve results using BERT-Base with feed-forward Neural Network and BERT-Base with CNN for Classification.

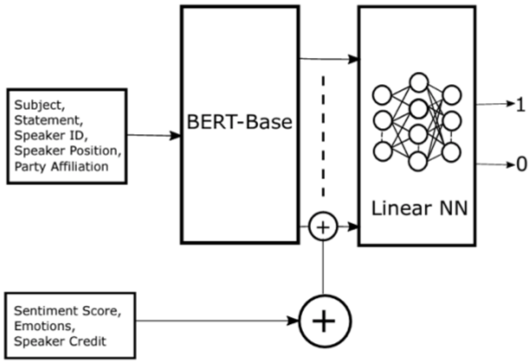


Fig. 2. BERT-Base with feed-forward component for classification

S.N.	Experiment	Accuracy	F1 Score Macro
1.	TEXT → [BB], BB_OP → [NN]	0.6882	0.5842
2.	TEXT+EMO → [BB], BB_OP → [NN]	0.6773	0.6352
3.	TEXT+EMO+ SPC → [BB], BB_OP → [NN]	0.6720	0.4021
4.	TEXT+EMO+ SPC+SEN → [BB], BB_OP → [NN]	0.6734	0.4097
5.	TEXT → [BB], BB_OP+EMP+ SPC+SEN → [NN]	0.6937	0.57234

TABLE II
BERT-BASE WITH FEED-FORWARD NN, ACCURACY AND F1 SCORE

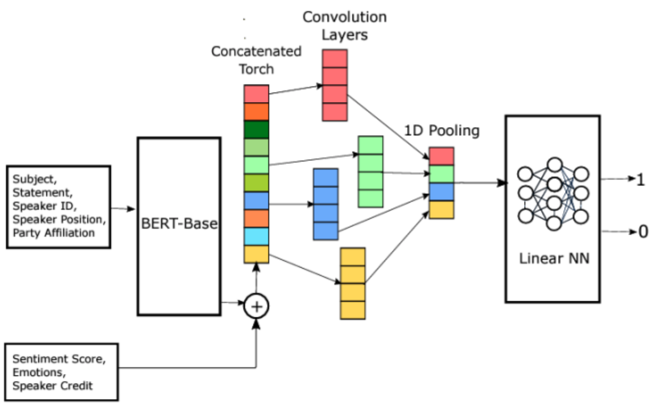


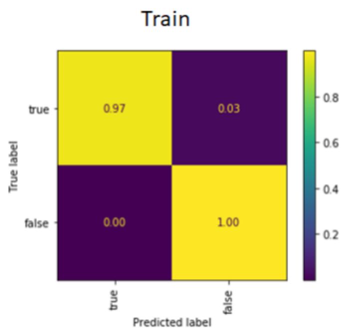
Fig. 3. BERT-Base with CNN for Classification

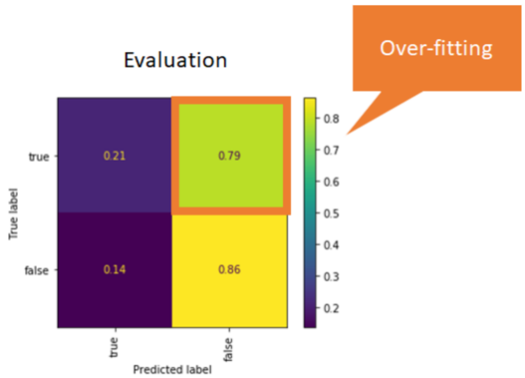
S.N.	Experiment	Accuracy	F1 Score Macro
1.	TEXT → [BB], BB_OP → [CNN]	0.6882	0.5308
2.	TEXT+EMO + SPC → [BB], BB_OP → [CNN]	0.5546	0.55641
3.	TEXT → [BB], BB_OP+ EMO → [CNN]	0.6554	0.608
4.	TEXT+SPC → [BB], BB_OP+ EMO → [CNN]	0.6890	0.6542
5.	TEXT → [BB], BB_OP+EMO +SPC → [CNN]	0.7000	0.6370
6.	TXT → [BB], BB_OP+EMO+ SPC+SEN → [CNN]	0.6992	0.6430

TABLE III
BERT-BASE + CNN, ACCURACY AND F1 SCORE

Improving on the result:

1. More baseline model:
   a. Multinomial Naïve Bayes
   b. Bernoulli Naïve Bayes
   c. Random Forest Classifier (fi-score: 0.58)

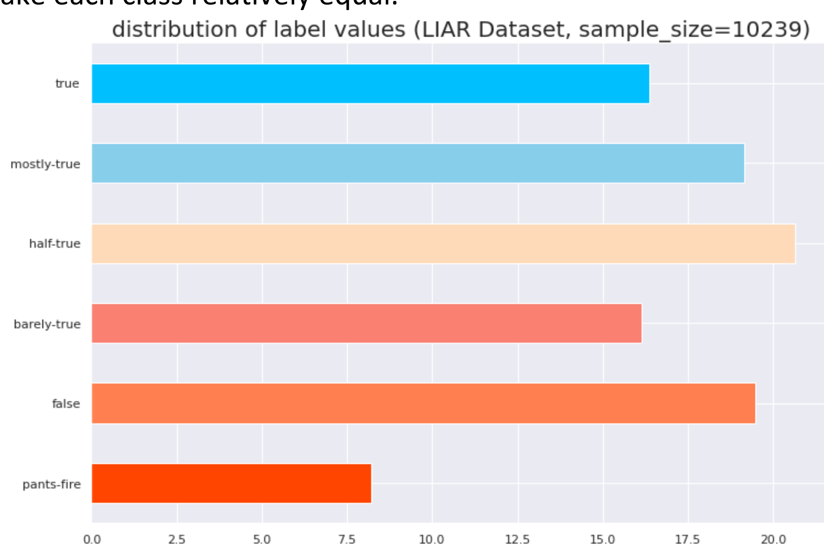




i. Over-fitting issue

d. Gradient boosting
e. RNN: recurrent neural networks

2. Oversampling/Undersampling for different class:
It is clearly that the distribution of label values are not well balanced, especially the pants-fire label, so we need to oversampling pants-fire or Undersampling other class to make each class relatively equal.



a. Bootstrap or Random Oversampling: Generating new samples by randomly sampling current data samples with substitution
b. SMOTE: Synthetic Minority Oversampling Technique is generating samples from the minority class using k-nearest neighbor method

c. ADASYN: Adaptive Synthetic Sampling Method, more synthetic data is produced for minority class samples that are more difficult to learn than for minority class samples that are simpler to learn.
d. NearMiss: When two points in the distribution belong to different classes and are extremely close to each other, this technique eliminates the data points from the larger class, attempting to balance the distribution

3. More advanced NLP tools: part of speech analysis (Hidden Markov Model and Viterbi Algorithm), google BERT model, Named Entity Recognition with Conditional Random Fields