MATH6380P: Final Project

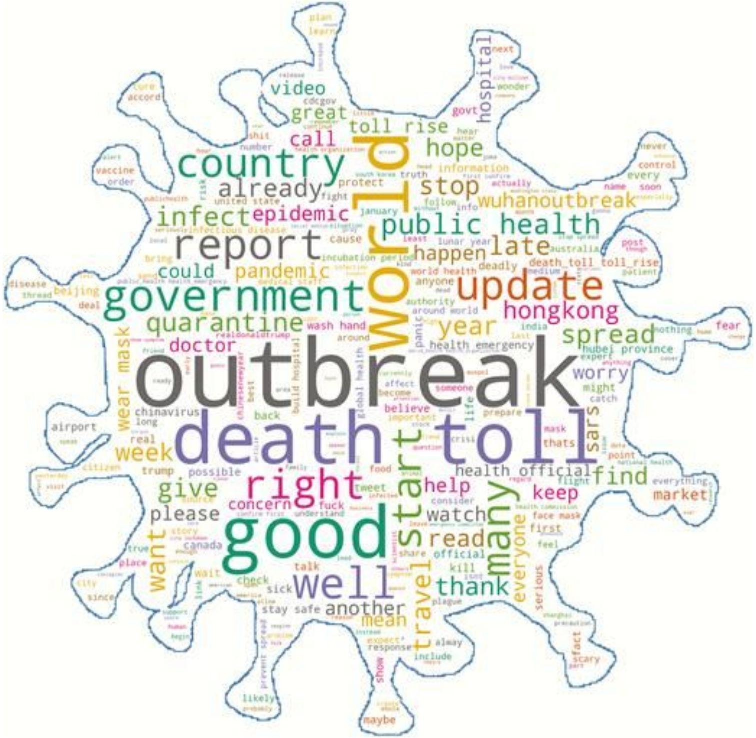
COVID-19 Fake-News Detection in Social Media Platforms

Ye Jin Bang, Etsuko Ishii , Samuel Cahyawijaya, Ziwei Ji
{yjbang, eishii, scahyawijaya, zjiad}@connect.ust.hk

COVID19 and ...

As the whole world has been going through a tough time due to the pandemic COVID-19, the amount of information about COVID-19 online grew exponentially and also spread rapidly across all social media platforms.

Suffering from...



Infodemic

overabundance of information,

a rapid and far-reaching spread of both accurate and inaccurate information about something

Suffering from Infodemic

NEWS

Hundreds dead' because of Covid-19 misinformation

By Alistair Coleman

CNN BUSINESS

'Fake news' about a Covid-19 vaccine has become a second pandemic, Red Cross chief says

Task

Task: To classify COVID-19 related social media posts into Real or Fake news.

Our aim is to achieve robust model for COVID-19 fake-news detection task through following two approaches:

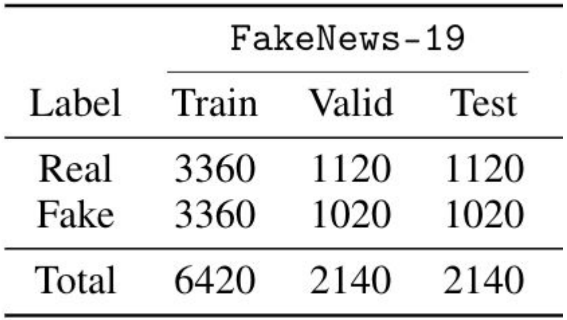
- Method 1: Training with robust loss functions;
- Method 2: Cleansing harmful training instances through influence calculation.

In addition, we also further evaluate adaptability of our method by performing zero-shot evaluation on different COVID-19 misinformation test set [2].

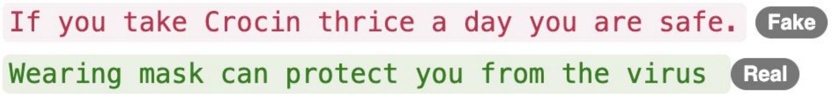
Dataset

FakeNews-19: consists of social media posts and articles of real and fake news on COVID19, all in English textual data.

- one of the subtasks on the hostile post detection proposed for Shared Task @CONSTRAINT 2021[1].
- Each of social media posts is manually annotated either as ``Fake'' or ``Real'', depending on its veracity



	FakeNews-19		
Label	Train	Valid	Test
Real	3360	1120	1120
Fake	3360	1020	1020
Total	6420	2140	2140



If you take Crocin thrice a day you are safe. Fake

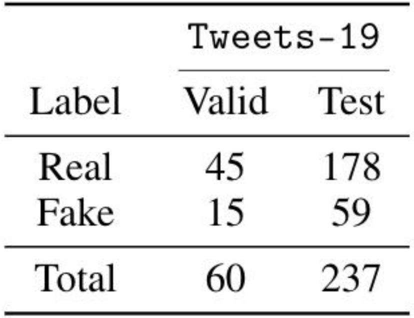
Wearing mask can protect you from the virus Real

[1] https://constraint-shared-task-2021.github.io/

Dataset

Tweets-19: Another COVID-19 infodemic tweets released by Alam et. al, for zero-shot test setting.

- Originally, labelled with several different questions for different usages.
- Taking Q2: *“To what extent does the tweet appear to contain false information?”*
- Map multi-labels to “Fake” or “Real” to incorporate with our binary setting



	Tweets-19	
Label	Valid	Test
Real	45	178
Fake	15	59
Total	60	237

[2] https://github.com/firojalam/COVID-19-tweets-for-check-worthiness

Method 1- Fine-tuning Pre-trained Transformer based Language Models with Robust Loss Functions

Feature Extractor: Transformers based language models

Model: Transformer based large-scale pre-trained language models with a feed-forward classifier trained on top of each model.

Training Method: Different Robust Loss functions

Method 1- Fine-tuning Pre-trained Transformer based Language Models with Robust Loss Functions

Robust Loss Functions:

- Symmetric Cross-Entropy (SCE) [1]
  - Inspired by the symmetric Kullback-Leibler divergence, SCE takes an additional term called reverse cross-entropy to enhance CE symmetricity.
- Generalized Cross-Entropy (GCE) [2]
  - GCE takes the advantages of both mean absolute error being noise-robust and CE performing well with challenging datasets.
- Curriculum Loss (CL) [3]
  - CL is a recently proposed 0-1 loss function which is a tighter upper bound compared with conventional summation based surrogate losses.

[1] Wang et al. "Symmetric cross entropy for robust learning with noisy labels." *ICCV 2019.*
[2] Ciavolino et al.. "Generalized cross entropy method for analysing the SERVQUAL model." J. *Appl. Stat. 42.3 (2015): 520-534.*
[3] Yueming et al. "Curriculum loss: Robust learning and generalization against label corruption." *arXiv:1905.10045* (2019).

Method 2- Cleansing harmful training instances through influence calculation

Data Noise Cleansing based on Training Instance Influence [1] estimates the influence of training instances given a target instance by introducing turn-over dropout mechanism. This approach is for achieving good performance and also a better generalization.

$$I(d^{\mathrm{tgt}}, d_i^{\mathrm{trn}}, f) = \mathcal{L}(f^{\widetilde{h(d_i^{\mathrm{trn}})}}, d^{\mathrm{tgt}}) - \mathcal{L}(f^{h(d_i^{\mathrm{trn}})}, d^{\mathrm{tgt}})$$

$$I_{\mathrm{tot}}(D^{\mathrm{tgt}}, d_i^{\mathrm{trn}}, f) = \sum_{j=1}^{K} I(d_j^{\mathrm{tgt}}, d_i^{\mathrm{trn}}, f)$$

[1] Kobayashi, et al. "Efficient Estimation of Influence of a Training Instance." *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. 2020.

Experiment 1:

Fine-tuning robust language models with robust loss functions

Experiment 1: Fine-tuning robust language models and loss functions

We setup the baseline of our experiment as a SVM model trained with features extracted from TF-IDF, provided by the dataset author.

We try with five different BERT-based models, including **ALBERT-base**, **BERT-base**, **BERT-large**, **RoBERTa-base**, and **RoBERTa-large**. We fine-tune the models with the classification layers on the top exploiting the pre-trained models.

We train each model with four different loss functions, which are **CE**, **SCE**, **GCE** and **CL**.

Hyperparameters:

- Learning rate: 1e-6, 3e-6, 5e-6
- Epoch: 1, 3, 5, 10

Experiment 1: Result on FakeNews-19

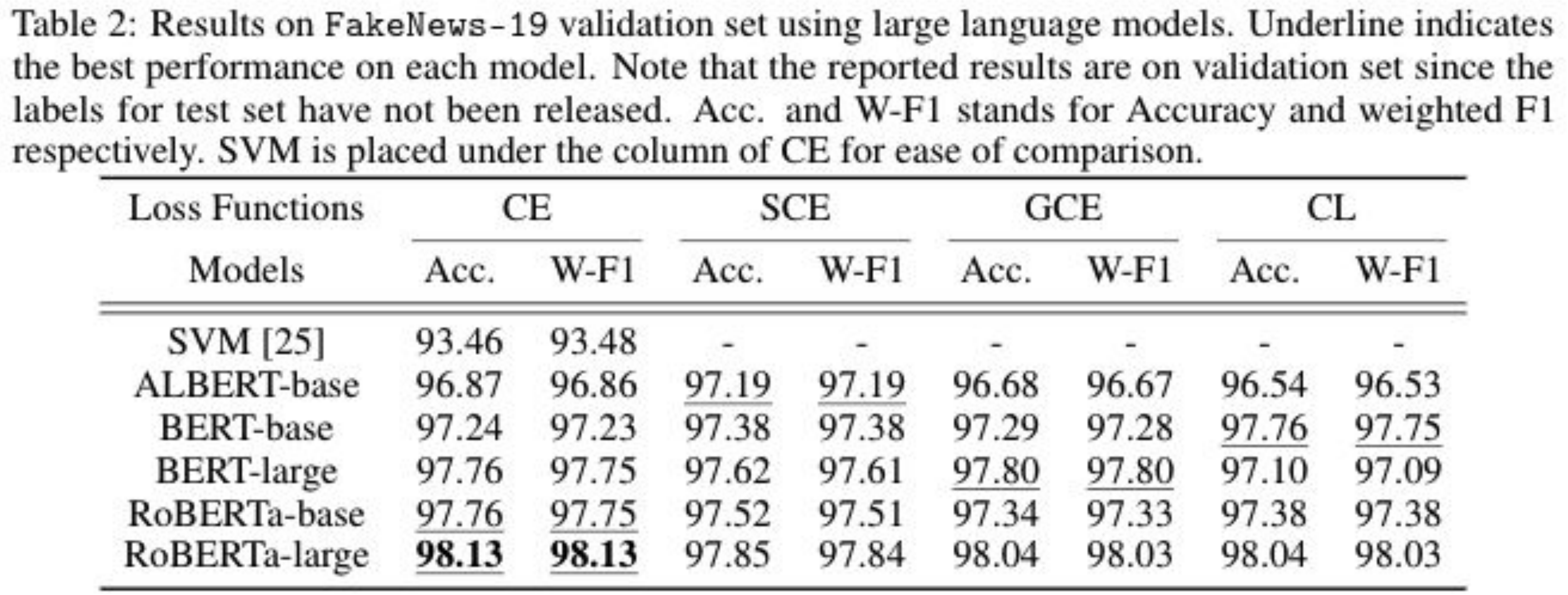


Table 2: Results on FakeNews-19 validation set using large language models. Underline indicates the best performance on each model. Note that the reported results are on validation set since the labels for test set have not been released. Acc. and W-F1 stands for Accuracy and weighted F1 respectively. SVM is placed under the column of CE for ease of comparison.

Loss Functions	CE		SCE		GCE		CL	
Models	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
SVM [25]	93.46	93.48	-	-	-	-	-	-
ALBERT-base	96.87	96.86	97.19	97.19	96.68	96.67	96.54	96.53
BERT-base	97.24	97.23	97.38	97.38	97.29	97.28	97.76	97.75
BERT-large	97.76	97.75	97.62	97.61	97.80	97.80	97.10	97.09
RoBERTa-base	97.76	97.75	97.52	97.51	97.34	97.33	97.38	97.38
RoBERTa-large	**98.13**	**98.13**	97.85	97.84	98.04	98.03	98.04	98.03

Experiment 1: Result on FakeNews-19

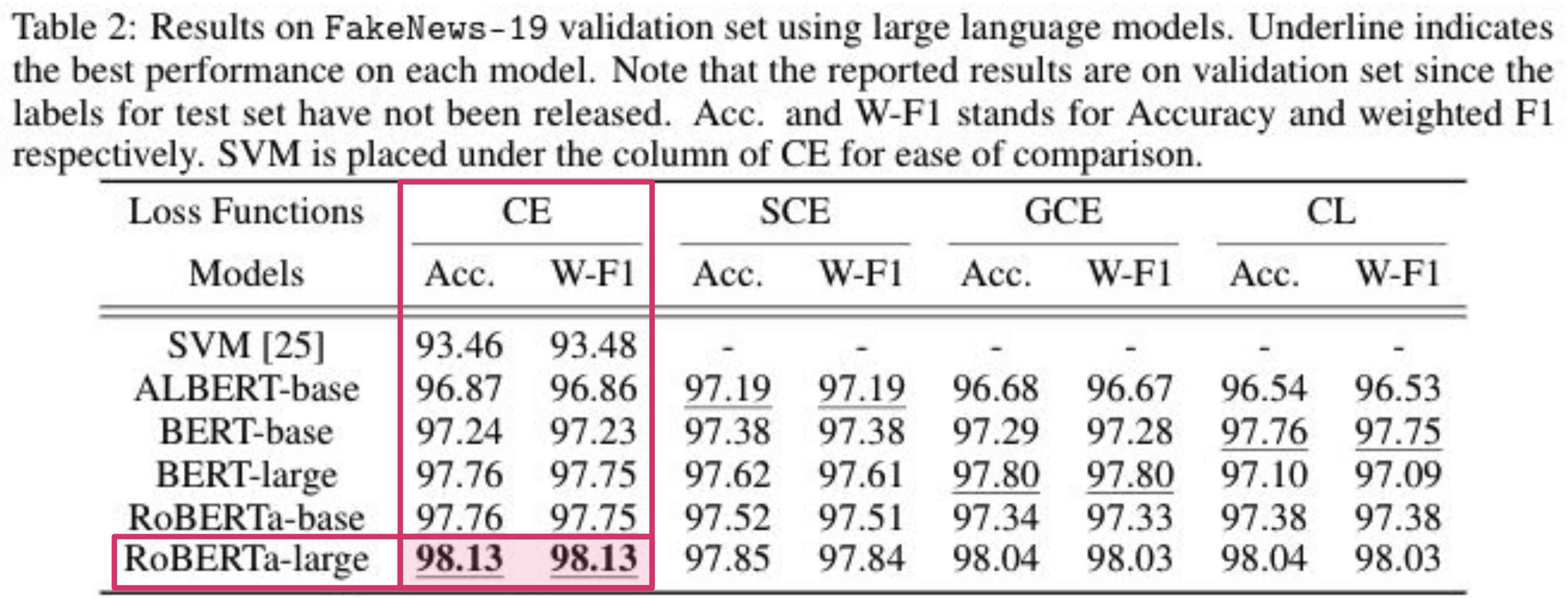


Table 2: Results on FakeNews-19 validation set using large language models. Underline indicates the best performance on each model. Note that the reported results are on validation set since the labels for test set have not been released. Acc. and W-F1 stands for Accuracy and weighted F1 respectively. SVM is placed under the column of CE for ease of comparison.

Loss Functions	CE		SCE		GCE		CL	
Models	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
SVM [25]	93.46	93.48	-	-	-	-	-	-
ALBERT-base	96.87	96.86	97.19	97.19	96.68	96.67	96.54	96.53
BERT-base	97.24	97.23	97.38	97.38	97.29	97.28	97.76	97.75
BERT-large	97.76	97.75	97.62	97.61	97.80	97.80	97.10	97.09
RoBERTa-base	97.76	97.75	97.52	97.51	97.34	97.33	97.38	97.38
RoBERTa-large	**98.13**	**98.13**	97.85	97.84	98.04	98.03	98.04	98.03

Experiment 1: Result on FakeNews-19

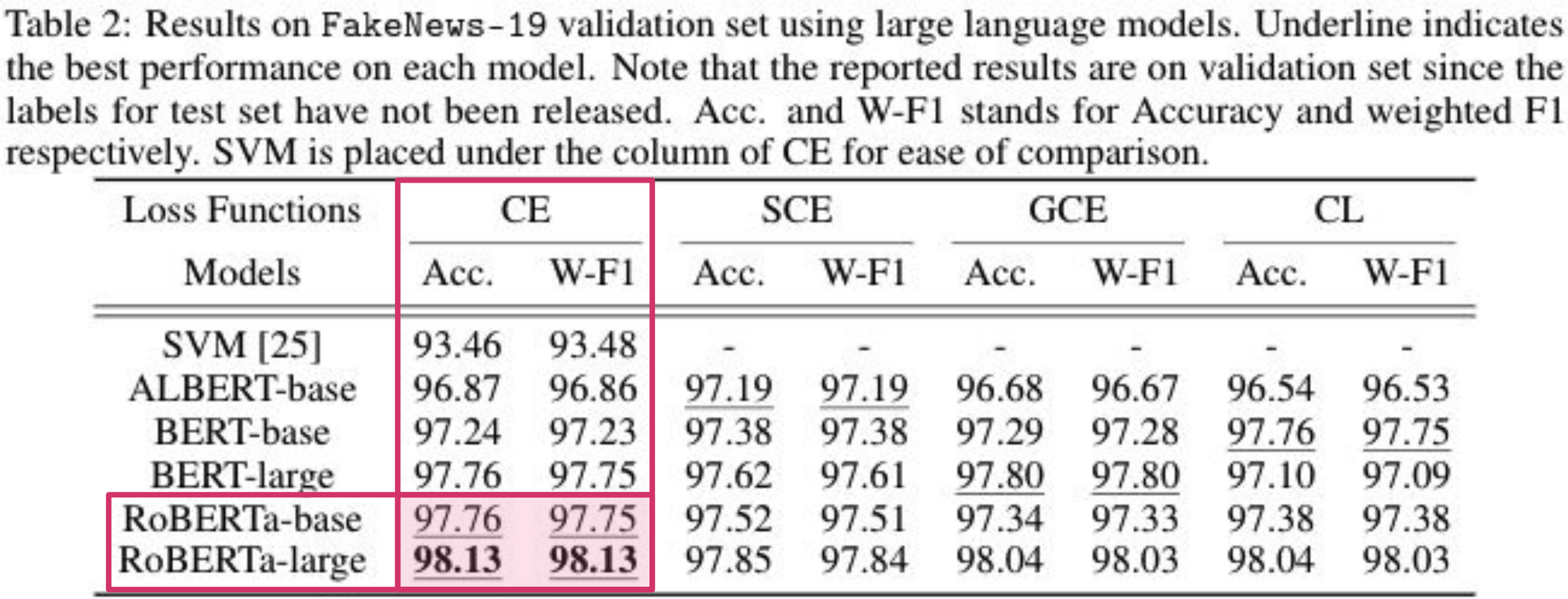


Table 2: Results on FakeNews-19 validation set using large language models. Underline indicates the best performance on each model. Note that the reported results are on validation set since the labels for test set have not been released. Acc. and W-F1 stands for Accuracy and weighted F1 respectively. SVM is placed under the column of CE for ease of comparison.

Loss Functions	CE		SCE		GCE		CL	
Models	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
SVM [25]	93.46	93.48	-	-	-	-	-	-
ALBERT-base	96.87	96.86	97.19	97.19	96.68	96.67	96.54	96.53
BERT-base	97.24	97.23	97.38	97.38	97.29	97.28	97.76	97.75
BERT-large	97.76	97.75	97.62	97.61	97.80	97.80	97.10	97.09
RoBERTa-base	97.76	97.75	97.52	97.51	97.34	97.33	97.38	97.38
RoBERTa-large	**98.13**	**98.13**	97.85	97.84	98.04	98.03	98.04	98.03

Experiment 1: Result on FakeNews-19

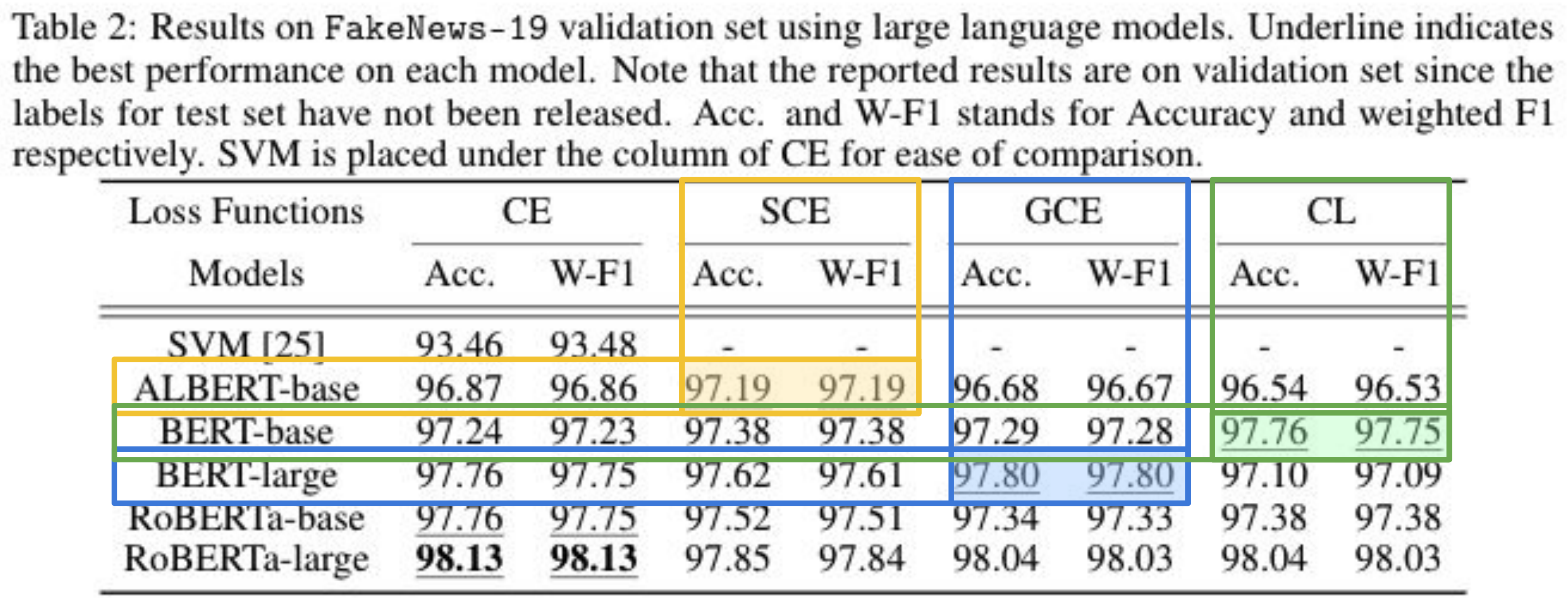


Table 2: Results on FakeNews-19 validation set using large language models. Underline indicates the best performance on each model. Note that the reported results are on validation set since the labels for test set have not been released. Acc. and W-F1 stands for Accuracy and weighted F1 respectively. SVM is placed under the column of CE for ease of comparison.

Loss Functions	CE		SCE		GCE		CL	
Models	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
SVM [25]	93.46	93.48	-	-	-	-	-	-
ALBERT-base	96.87	96.86	97.19	97.19	96.68	96.67	96.54	96.53
BERT-base	97.24	97.23	97.38	97.38	97.29	97.28	97.76	97.75
BERT-large	97.76	97.75	97.62	97.61	97.80	97.80	97.10	97.09
RoBERTa-base	97.76	97.75	97.52	97.51	97.34	97.33	97.38	97.38
RoBERTa-large	**98.13**	**98.13**	97.85	97.84	98.04	98.03	98.04	98.03

Experiment 1: Result on FakeNews-19

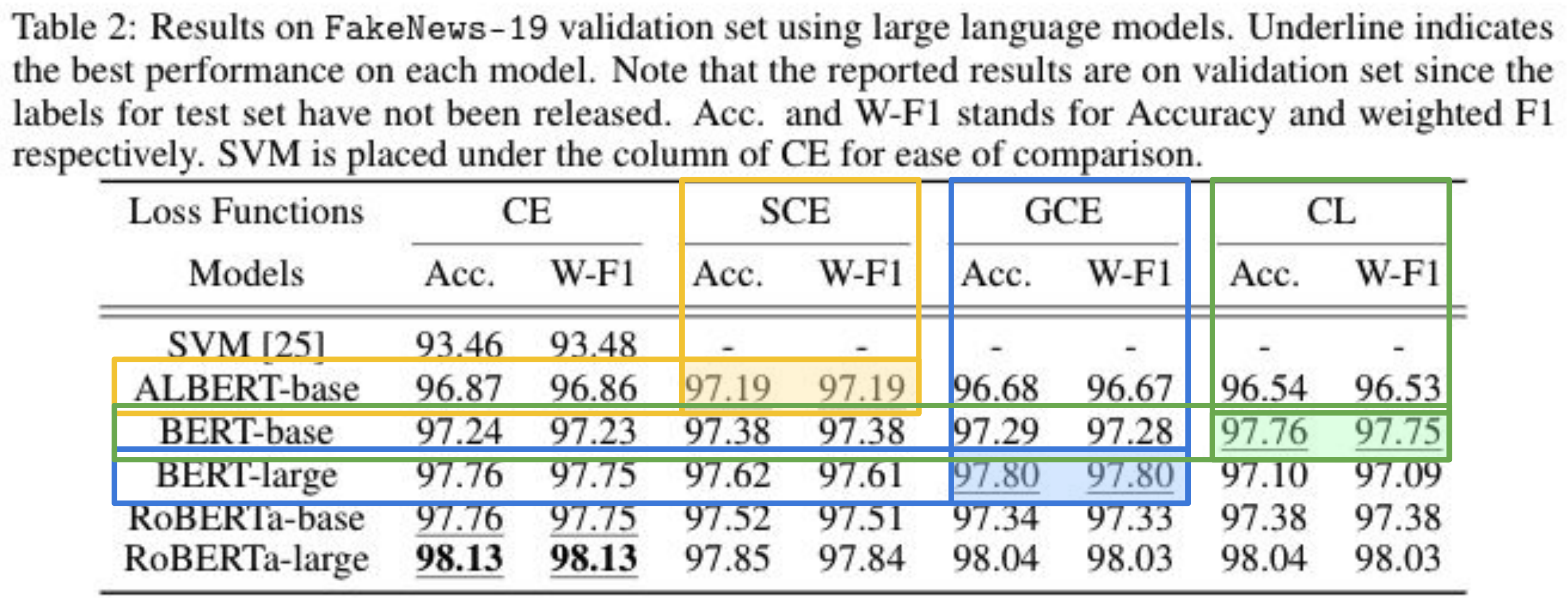


Table 2: Results on FakeNews-19 validation set using large language models. Underline indicates the best performance on each model. Note that the reported results are on validation set since the labels for test set have not been released. Acc. and W-F1 stands for Accuracy and weighted F1 respectively. SVM is placed under the column of CE for ease of comparison.

Loss Functions	CE		SCE		GCE		CL	
Models	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
SVM [25]	93.46	93.48	-	-	-	-	-	-
ALBERT-base	96.87	96.86	97.19	97.19	96.68	96.67	96.54	96.53
BERT-base	97.24	97.23	97.38	97.38	97.29	97.28	97.76	97.75
BERT-large	97.76	97.75	97.62	97.61	97.80	97.80	97.10	97.09
RoBERTa-base	97.76	97.75	97.52	97.51	97.34	97.33	97.38	97.38
RoBERTa-large	**98.13**	**98.13**	97.85	97.84	98.04	98.03	98.04	98.03

Experiment 1: Result on **FakeNews-19**

- RoBERTa-large trained with cross entropy loss function performs the best in both F1 and accuracy scores.
- RoBERTa extracts features well enough that robust loss functions can merely contribute
- ALBERT and BERT models slightly improve thanks to the robust loss functions

Experiment 1: Result on **Tweets-19**

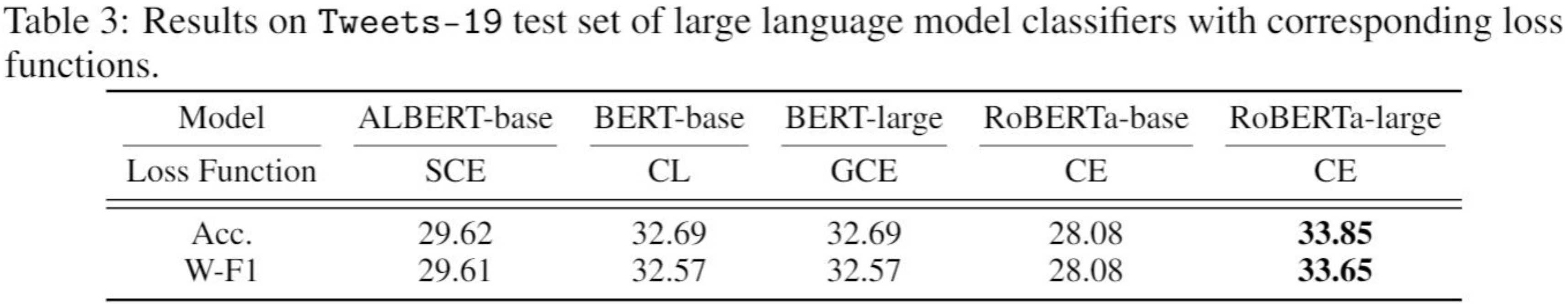
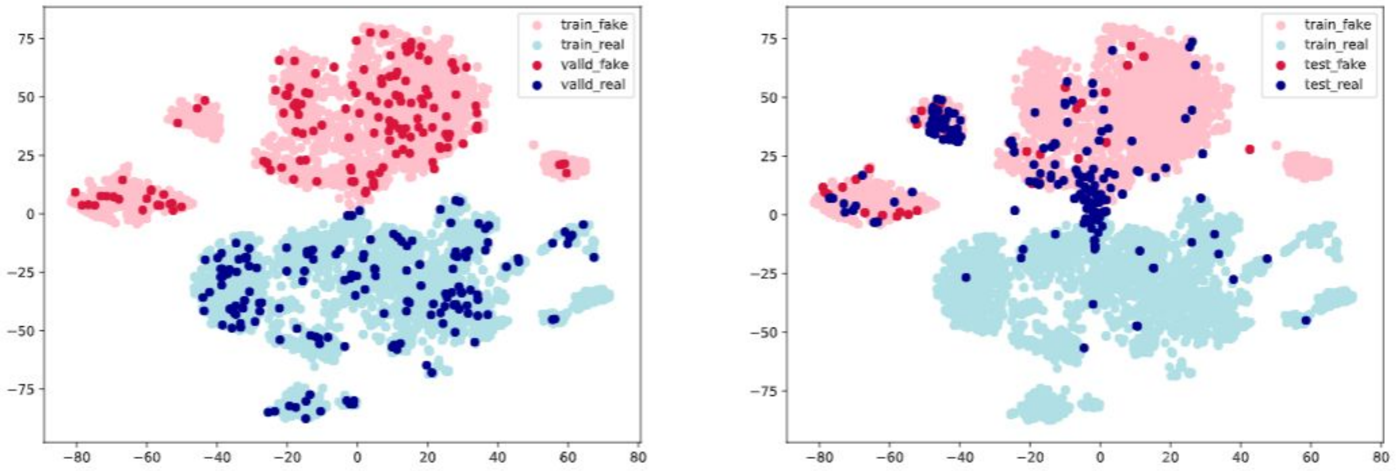


Table 3: Results on `Tweets-19` test set of large language model classifiers with corresponding loss functions.

Model	ALBERT-base	BERT-base	BERT-large	RoBERTa-base	RoBERTa-large
Loss Function	SCE	CL	GCE	CE	CE
Acc.	29.62	32.69	32.69	28.08	**33.85**
W-F1	29.61	32.57	32.57	28.08	**33.65**

Analysis 1: Data distribution of two Testsets



(a) FakeNews-19 validation set.

(b) Tweets-19 test set.

Figure 1: Datasets distribution comparison with FakeNews-19 training set using t-SNE. While the distributions within FakeNews-19 kept to be similar, the distribution of Tweets-19 is significantly different.

Experiment 2:

Zero-shot with Data Cleansing

Experiment 2: Zero-shot with Data Cleansing

We test the adaptability of our data cleansing method by performing zero-shot evaluation on **Tweet-19** dataset.

We first fine-tune a pre-trained RoBERTa-large model with **FakeNews-19** while applying **turn-over dropout** to the weight matrix on the last affine transformation layer of the model with dropout probability of **0.5**.

We calculate the **total influence score** from the validation set of **Tweets-19**. We investigate the effectiveness of our data cleansing approach by removing n% of training instances with smallest total influence score with **n = { 1, 5, 10, 50, 75, 90, 95, 99}** and retrain the models the remaining training data.

All models are trained with **Cross Entropy** and learning rate of **3e-6**

Experiment 2: Result

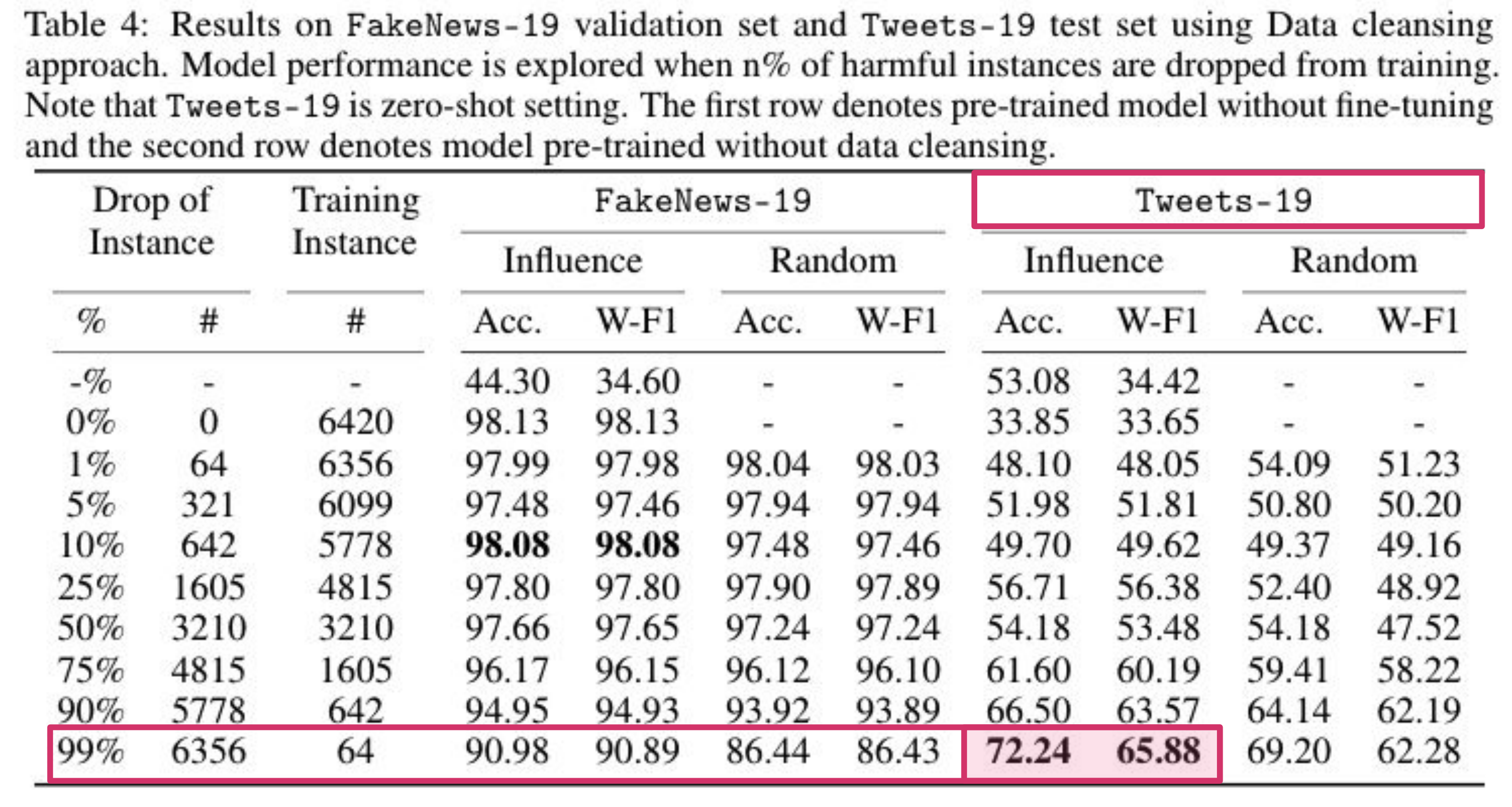


Table 4: Results on FakeNews-19 validation set and Tweets-19 test set using Data cleansing approach. Model performance is explored when n% of harmful instances are dropped from training. Note that Tweets-19 is zero-shot setting. The first row denotes pre-trained model without fine-tuning and the second row denotes model pre-trained without data cleansing.

Drop of Instance		Training Instance	FakeNews-19				Tweets-19			
			Influence		Random		Influence		Random	
%	#	#	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
-%	-	-	44.30	34.60	-	-	53.08	34.42	-	-
0%	0	6420	98.13	98.13	-	-	33.85	33.65	-	-
1%	64	6356	97.99	97.98	98.04	98.03	48.10	48.05	54.09	51.23
5%	321	6099	97.48	97.46	97.94	97.94	51.98	51.81	50.80	50.20
10%	642	5778	**98.08**	**98.08**	97.48	97.46	49.70	49.62	49.37	49.16
25%	1605	4815	97.80	97.80	97.90	97.89	56.71	56.38	52.40	48.92
50%	3210	3210	97.66	97.65	97.24	97.24	54.18	53.48	54.18	47.52
75%	4815	1605	96.17	96.15	96.12	96.10	61.60	60.19	59.41	58.22
90%	5778	642	94.95	94.93	93.92	93.89	66.50	63.57	64.14	62.19
99%	6356	64	90.98	90.89	86.44	86.43	**72.24**	**65.88**	69.20	62.28

Experiment 2: Result

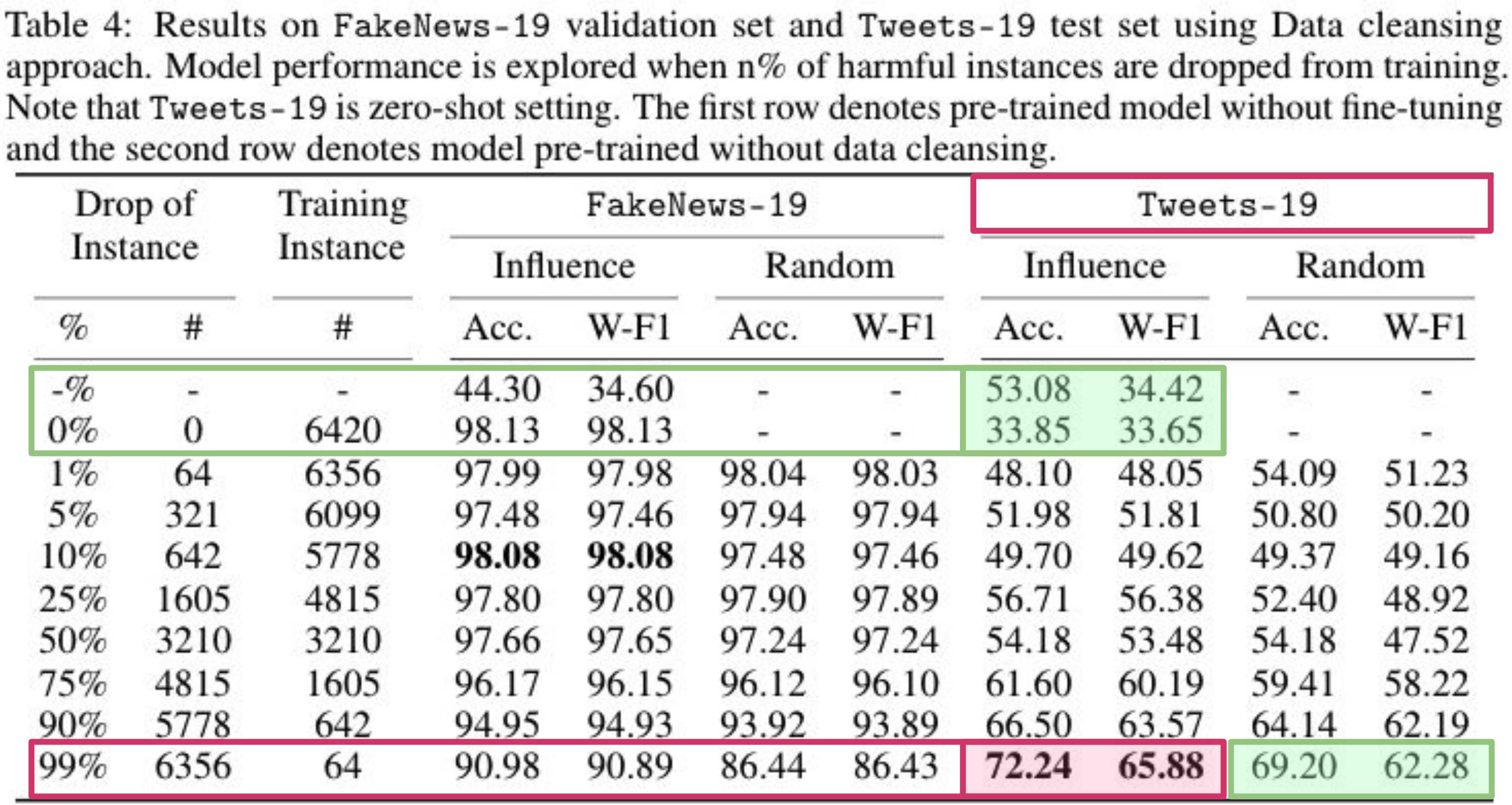
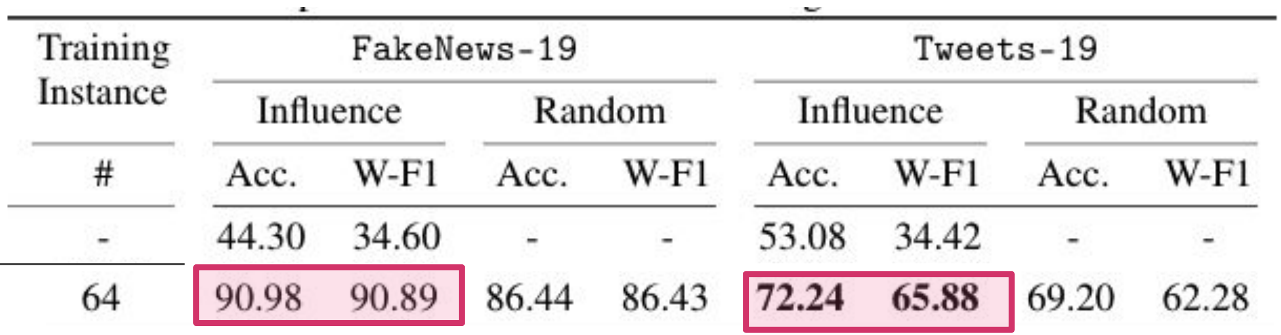


Table 4: Results on FakeNews-19 validation set and Tweets-19 test set using Data cleansing approach. Model performance is explored when n% of harmful instances are dropped from training. Note that Tweets-19 is zero-shot setting. The first row denotes pre-trained model without fine-tuning and the second row denotes model pre-trained without data cleansing.

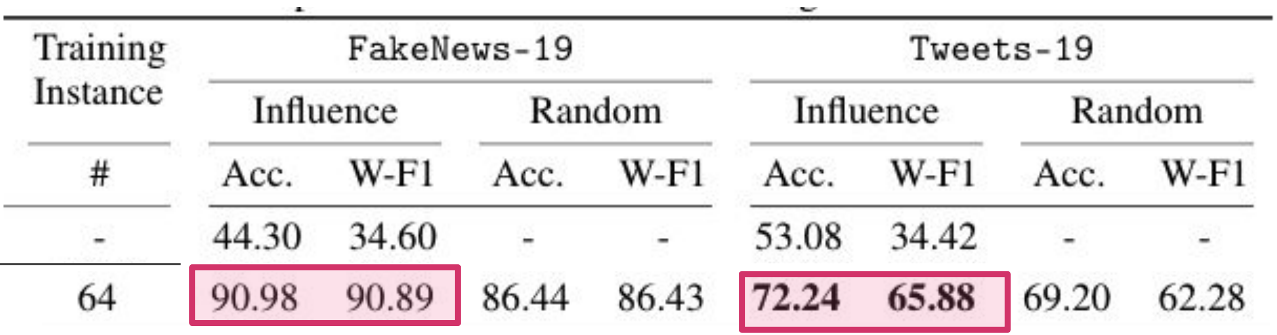
Drop of Instance		Training Instance	FakeNews-19				Tweets-19			
			Influence		Random		Influence		Random	
%	#	#	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
-%	-	-	44.30	34.60	-	-	53.08	34.42	-	-
0%	0	6420	98.13	98.13	-	-	33.85	33.65	-	-
1%	64	6356	97.99	97.98	98.04	98.03	48.10	48.05	54.09	51.23
5%	321	6099	97.48	97.46	97.94	97.94	51.98	51.81	50.80	50.20
10%	642	5778	**98.08**	**98.08**	97.48	97.46	49.70	49.62	49.37	49.16
25%	1605	4815	97.80	97.80	97.90	97.89	56.71	56.38	52.40	48.92
50%	3210	3210	97.66	97.65	97.24	97.24	54.18	53.48	54.18	47.52
75%	4815	1605	96.17	96.15	96.12	96.10	61.60	60.19	59.41	58.22
90%	5778	642	94.95	94.93	93.92	93.89	66.50	63.57	64.14	62.19
99%	6356	64	90.98	90.89	86.44	86.43	**72.24**	**65.88**	69.20	62.28

Experiment 2: Result



	Training Instance	FakeNews-19				Tweets-19			
		Influence		Random		Influence		Random	
	#	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
Without Fine-tuning	-	44.30	34.60	-	-	53.08	34.42	-	-
With Fine-tuning	64	90.98	90.89	86.44	86.43	**72.24**	**65.88**	69.20	62.28

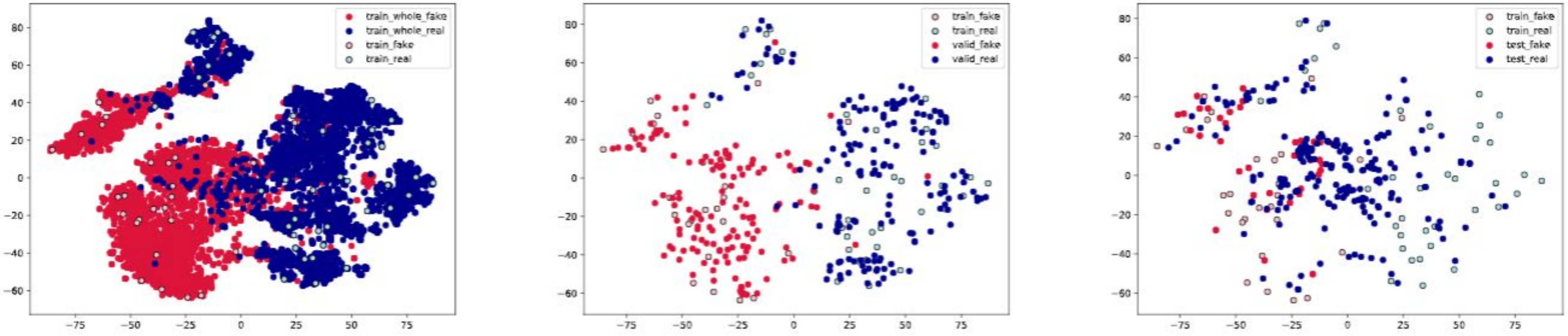
Experiment 2: Result



	Training Instance	FakeNews-19				Tweets-19			
		Influence		Random		Influence		Random	
	#	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1	Acc.	W-F1
Without Fine-tuning	-	44.30	34.60	-	-	53.08	34.42	-	-
With Fine-tuning	64	90.98	90.89	86.44	86.43	**72.24**	**65.88**	69.20	62.28

- The 64 most influential training instances significantly boost generalization of the model

Analysis 2: Why smaller data helped for generalization?



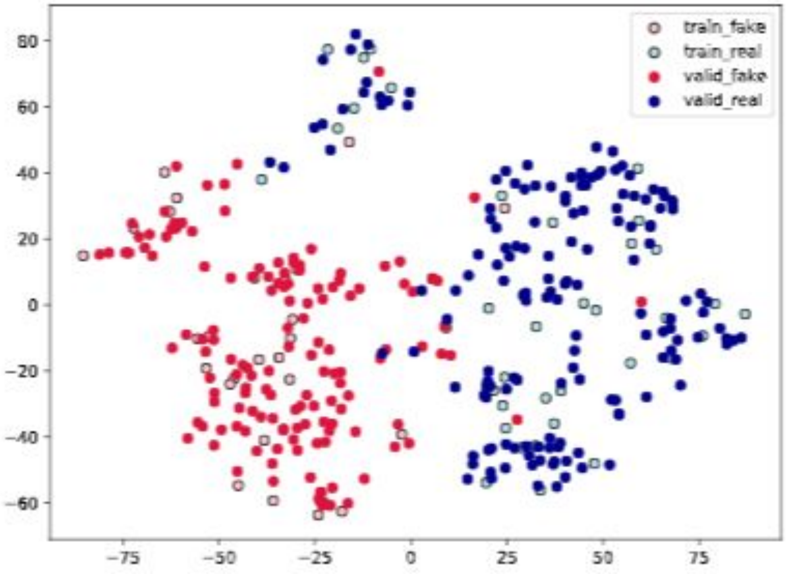
(a) Whole FakeNews-19 train set.

(b) FakeNews-19 validation set.

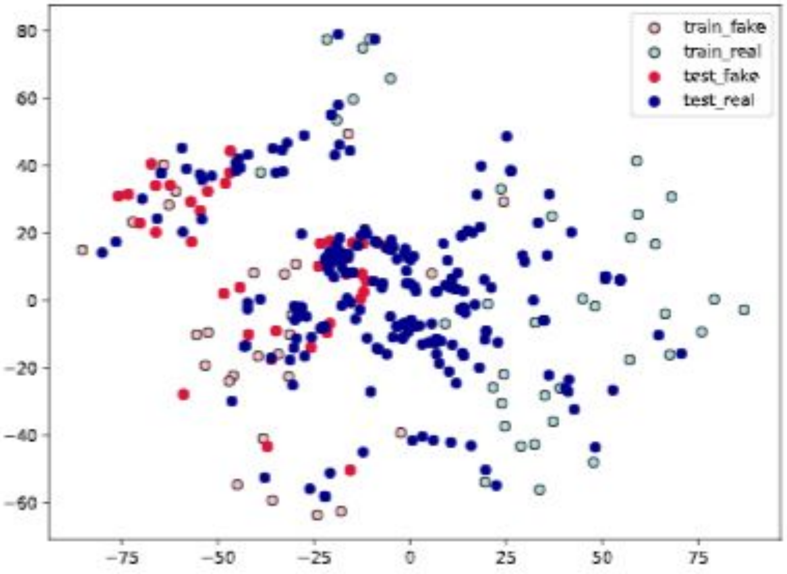
(c) Tweets-19 test set.

Figure 2: Datasets distribution comparison with top 1% influential training samples using t-SNE. Top 1% influential samples are distributed fairly evenly over the whole training set (a), thus the extracted validation features remain separable (b), and the Tweets-19 distribution is captured better than trained with the full training set (c).

Analysis 2: Why smaller data helped for generalization?



(b) FakeNews-19 validation set.



(c) Tweets-19 test set.

Analysis 3:

RoBERTa feature extraction + Traditional Classifiers

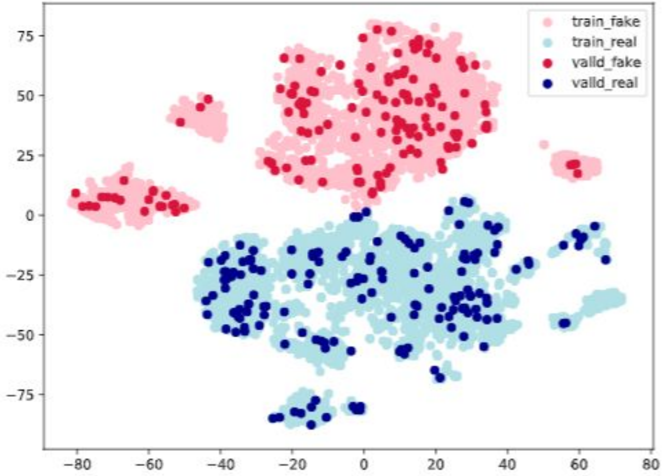
Analysis 3: RoBERTa feature extraction + Traditional Classifiers

Motivation:

- The baseline with TF-IDF feature extraction with SVM achieves somewhat satisfactory evaluation performance.
- The features extracted by the fine-tuned RoBERTa seem linearly separable

Experiments: replacing NN to traditional classifiers

1. logistic regression
2. SVM
3. gradient boosting



(a) FakeNews-19 validation set.

Analysis 3: RoBERTa feature extraction + Traditional Classifiers

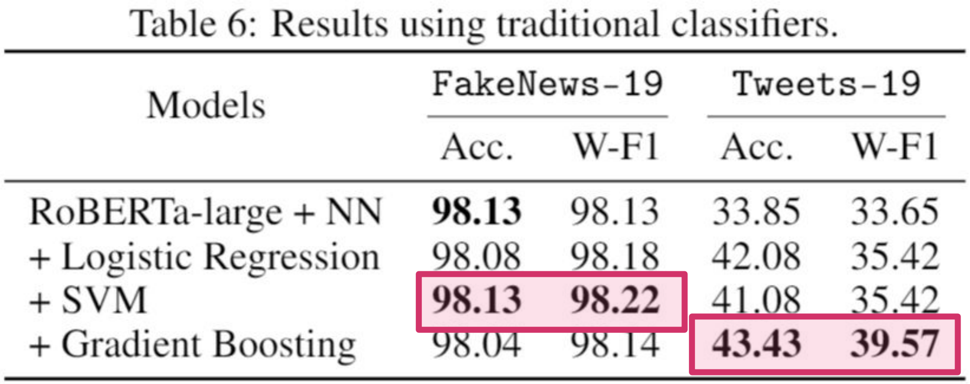


Table 6: Results using traditional classifiers.

Models	FakeNews-19		Tweets-19	
	Acc.	W-F1	Acc.	W-F1
RoBERTa-large + NN	**98.13**	98.13	33.85	33.65
+ Logistic Regression	98.08	98.18	42.08	35.42
+ SVM	**98.13**	**98.22**	41.08	35.42
+ Gradient Boosting	98.04	98.14	**43.43**	**39.57**

- Perform as good as the RoBERTa model on FakeNews-19, and even outperform on Tweets-19.

Analysis 3: RoBERTa feature extraction + Traditional Classifiers

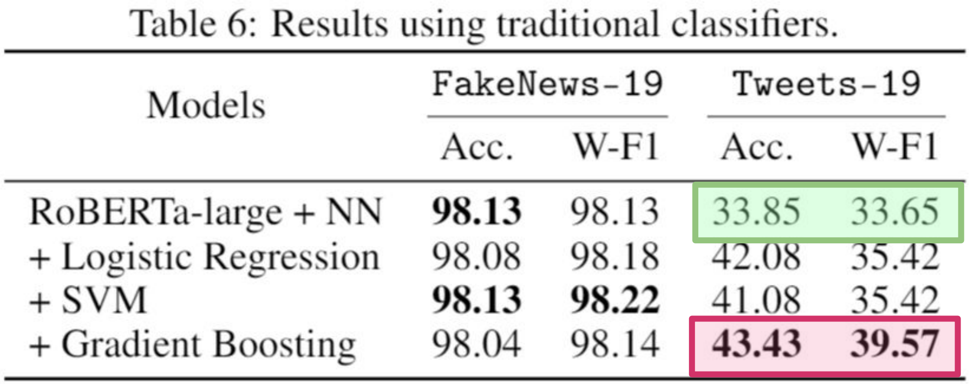


Table 6: Results using traditional classifiers.

Models	FakeNews-19		Tweets-19	
	Acc.	W-F1	Acc.	W-F1
RoBERTa-large + NN	**98.13**	98.13	33.85	33.65
+ Logistic Regression	98.08	98.18	42.08	35.42
+ SVM	**98.13**	**98.22**	41.08	35.42
+ Gradient Boosting	98.04	98.14	**43.43**	**39.57**

- Perform as good as the RoBERTa model on FakeNews-19, and even outperform on Tweets-19.

- NN classifier has too many parameters thus causing overfitting to FakeNews-19.

Analysis 3: RoBERTa feature extraction + Traditional Classifiers

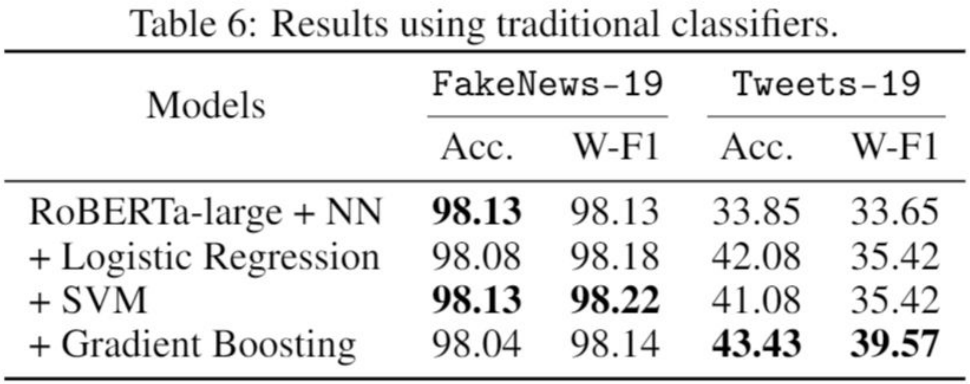


Table 6: Results using traditional classifiers.

Models	FakeNews-19		Tweets-19	
	Acc.	W-F1	Acc.	W-F1
RoBERTa-large + NN	**98.13**	98.13	33.85	33.65
+ Logistic Regression	98.08	98.18	42.08	35.42
+ SVM	**98.13**	**98.22**	41.08	35.42
+ Gradient Boosting	98.04	98.14	**43.43**	**39.57**

- Perform as good as the RoBERTa model on FakeNews-19, and even outperform on Tweets-19.
- NN classifier has too many parameters thus causing overfitting to FakeNews-19.

Traditional algorithms is preferable in terms of <u>computational efficiency</u> and <u>model transparency</u> while it does not harm the performance.

Conclusion

Conclusion

In this project, we explored COVID-19 fake news detection problem.

We fine-tuned different pre-trained large language models with different robust loss functions and tested on in-domain and out-domain dataset to see the generalization of the model.

We figure out that training with robust loss function doesn't really help the model to generalize better and only achieve evaluation performance of 33.85% accuracy and 33.65% W-F1 for **Tweets-19**.

Conclusion

We further explored influence-based data cleansing technique and with 99% cleansing percentage, our model can produce the best evaluation performance on `Tweets-19` with 72.24% accuracy score and 65.88% W-F1 score while still maintaining high enough validation performance on `FakeNews-19`.

Future Work

- Combine the robust loss function with the influence score data cleansing method
- The resulting influence score can be made more robust for removing outlier data in multiple domains setting.

Thank you.

Reference and Code available at
https://github.com/yjbang/math6380