Spoiler Classification in Book Reviews

Team Members: Rohan Saraogi, Shikha Reddy, Akash Sundar

Problem Formulation

- Sentence-level binary classification problem
- Given the raw text of a sentence in a review, predict whether it contains spoilers or not

Fine-Grained Spoiler Detection from Large-Scale Review Corpora

Mengting Wan[*], **Rishabh Misra**[†], **Ndapa Nakashole**[*], **Julian McAuley**[*]
[*]University of California, San Diego, [†]Amazon.com, Inc
{m5wan, r1misra, nnakashole, jmcauley}@ucsd.edu

- Scraped reviews from goodreads.com
- Spoiler tags are sentence-specific and self-reported by review authors

	user_id	timestamp	review_sentences	rating	has_spoiler	book_id	review_id
0	8842281e1d1347389f2ab93d60773d4d	2017-08-30	[[0, This is a special book.], [0, It started ...	5	True	18245960	dfdbb7b0eb5a7e4c26d59a937e2e5feb
1	8842281e1d1347389f2ab93d60773d4d	2017-03-22	[[0, Recommended by Don Katz.], [0, Avail for ...	3	False	16981	a5d2c3628987712d0e05c4f90798eb67
2	8842281e1d1347389f2ab93d60773d4d	2017-03-20	[[0, A fun, fast paced science fiction thrille...	3	True	28684704	2ede853b14dc4583f96cf5d120af636f
3	8842281e1d1347389f2ab93d60773d4d	2016-11-09	[[0, Recommended reading to understand what is...	0	False	27161156	ced5675e55cd9d38a524743f5c40996e
4	8842281e1d1347389f2ab93d60773d4d	2016-04-25	[[0, I really enjoyed this book, and there is ...	4	True	25884323	332732725863131279a8e345b63ac33e

Case	Value
No. of reviews	1,378,033
No. of users	18,892
No. of books	25,475
No. of sentences	17,672,655
% of spoiler sentences	3.22%

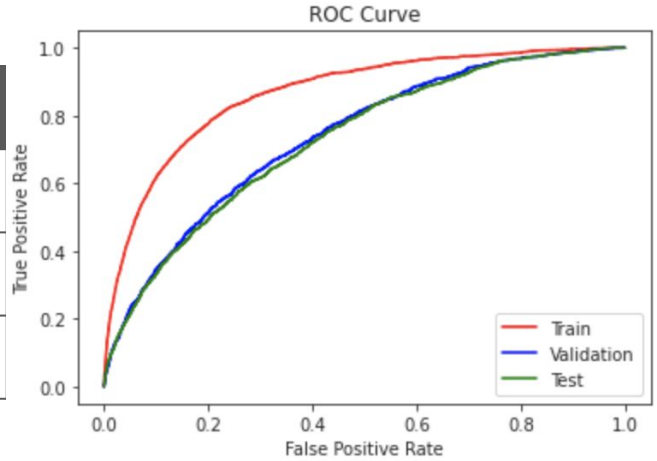
- Training Set: ~100,000
- Validation Set: 50,000
- Test Set: 50,000

Baseline

```
vectorizer = TfidfVectorizer(analyzer="word", max_features=20000)
```

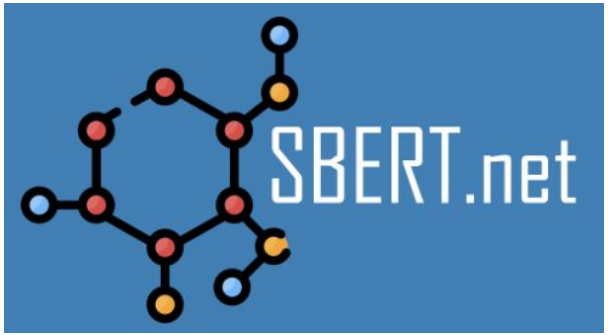
```
clf = LogisticRegression(penalty="l2", random_state=42, solver="lbfgs", max_iter=1000, verbose=1)
```

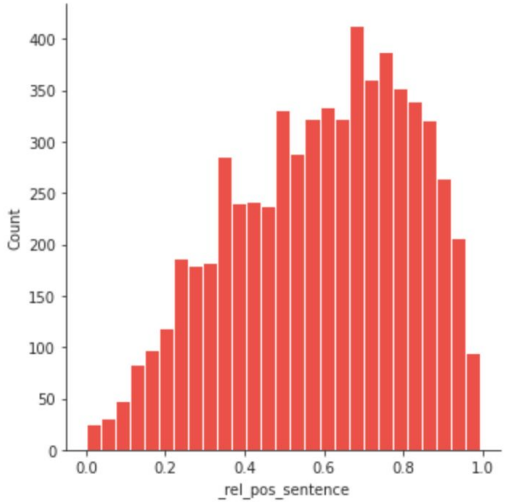
Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.967	nan	0.00	nan	0.865
Validation	0.970	1.00	0.00	0.00	0.738
Test	0.969	0.50	0.00	0.00	0.729

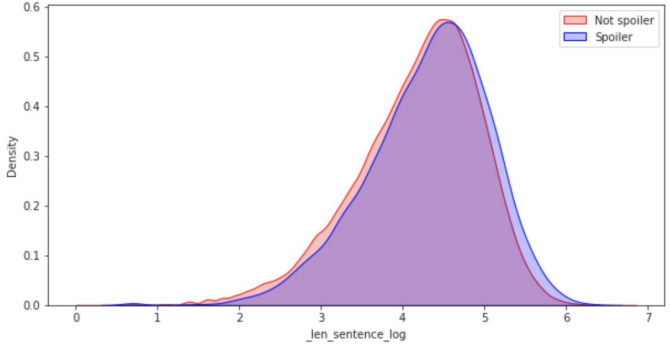


Feature Engineering

Sentence BERT (384,) + Relative position (1,) + Log of length (1,)







Class Imbalance

- Class weights

```
{"class_weight": [{0: w, 1: 1} for w in [0.01, 0.05, 0.1, 0.3, 0.5, 1]]}
```

- Undersampling majority class

```
{"sampling_strategy": [0.05, 0.1, 0.3, 0.5, 1]} # Undersampling majority
```

- Oversampling minority class (SMOTE)

```
{"sampling_strategy": [0.05, 0.1, 0.3, 0.5, 1]} # Oversampling minority
```

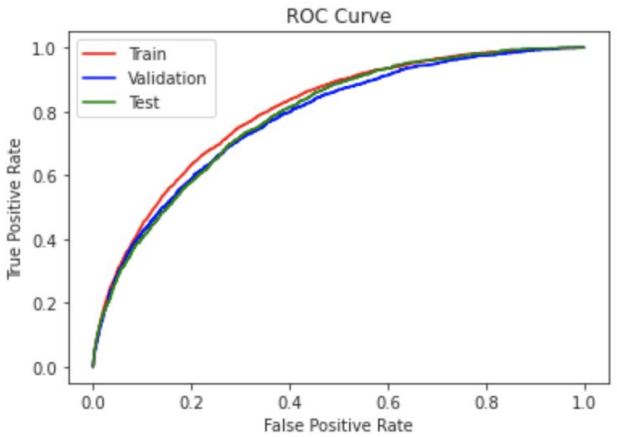
Logistic Regression

```
{"penalty": ["elasticnet"], "l1_ratio": [0, 0.5, 1]}
```

{ Class Weights OR Undersampling Majority OR Oversampling Minority }

l1_ratio = 0.5, class_weight = {0: 0.1, 1: 1}

Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.917	0.16	0.35	0.21	0.801
Validation	0.917	0.14	0.34	0.20	0.779
Test	0.918	0.14	0.32	0.19	0.785



Multi-Layer Perceptron Classifier

```
{'hidden_layer_sizes': [(128, 64, 32, 1)], 'activation': ['tanh', 'relu', 'logistic'], 'alpha': [0.0001, 0.05]}
```

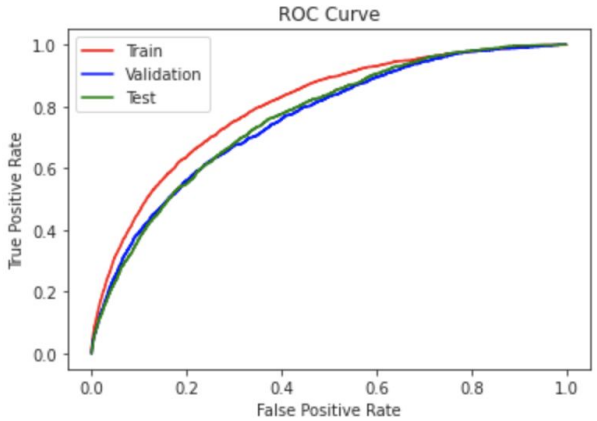
+

{ None OR Undersampling Majority OR Oversampling Minority }

activation="tanh", alpha=0.05, sampling_strategy=0.3***

Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.904	0.15	0.40	0.22	0.802
Validation	0.902	0.12	0.35	0.18	0.757
Test	0.901	0.11	0.32	0.16	0.762

*** Convergence issues



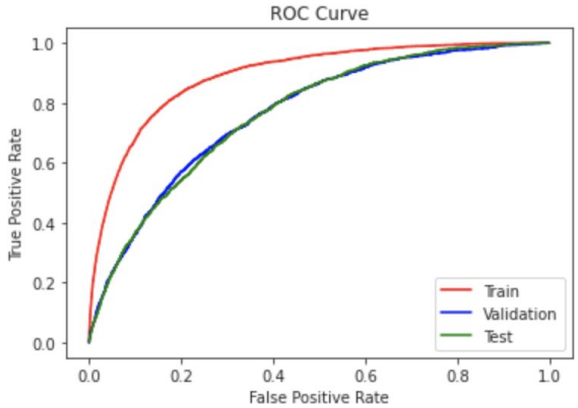
Random Forest Classifier

```
{"n_estimators": [100, 150], "max_depth": [5, 7], "min_samples_leaf": [1, 5]}
```

{ Class Weights OR Undersampling Majority OR Oversampling Minority }

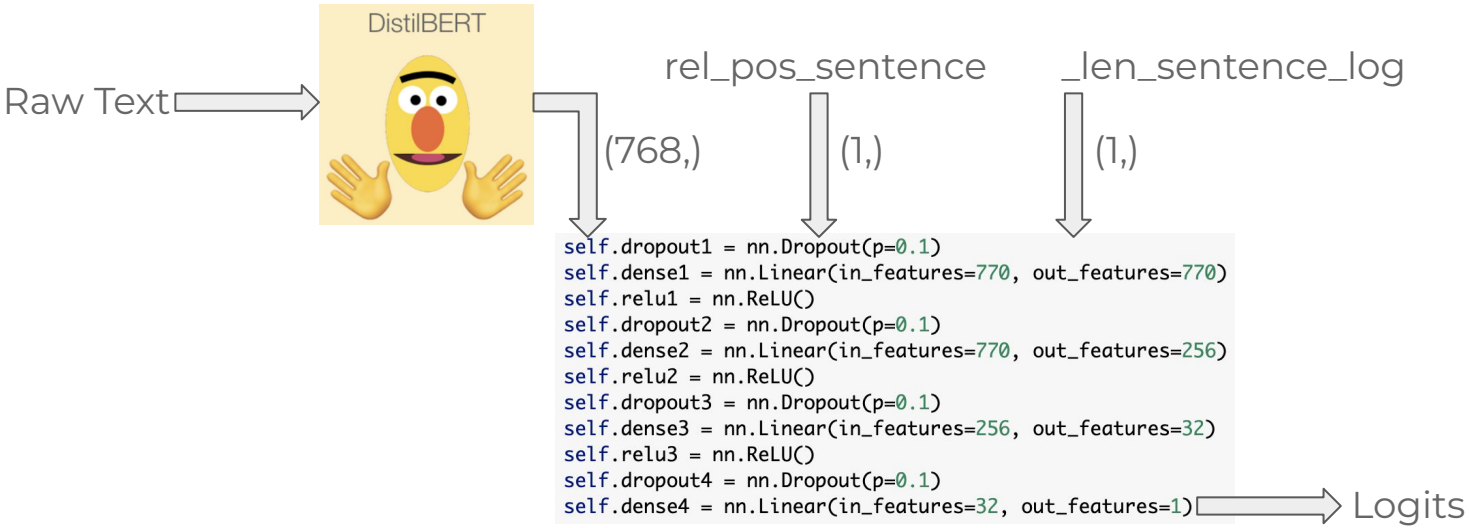
n_estimators=150, max_depth=7, min_samples_leaf=1, class_weight={0: 0.05, 1: 1}

Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.941	0.27	0.47	0.35	0.893
Validation	0.933	0.13	0.22	0.16	0.767
Test	0.932	0.13	0.22	0.16	0.766



Transfer Learning

Model Architecture



Loss Functions

- Binary Cross-Entropy Loss

$$l_n = -w_n \left[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n)) \right]$$

- Weighted Binary Cross-Entropy Loss

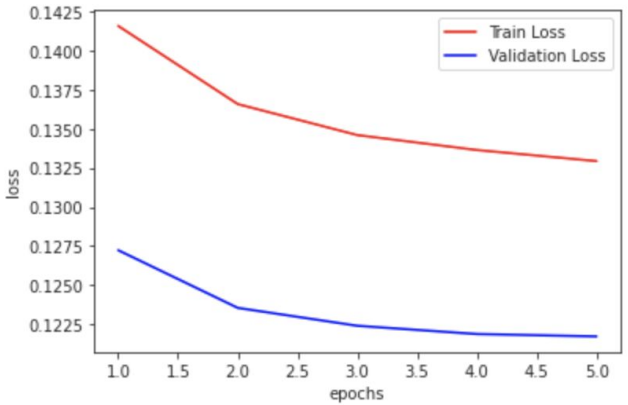
$$-w_{n,c} \left[p_c y_{n,c} \cdot \log \sigma(x_{n,c}) + (1 - y_{n,c}) \cdot \log(1 - \sigma(x_{n,c})) \right]$$

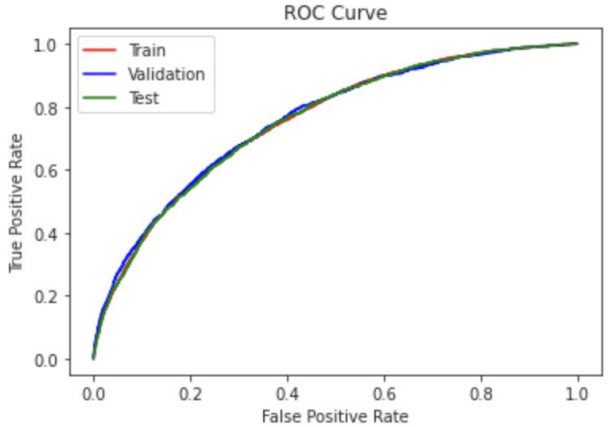
- Focal Loss

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases} \qquad \mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t)$$

Binary Cross-Entropy Loss (epochs = 5)

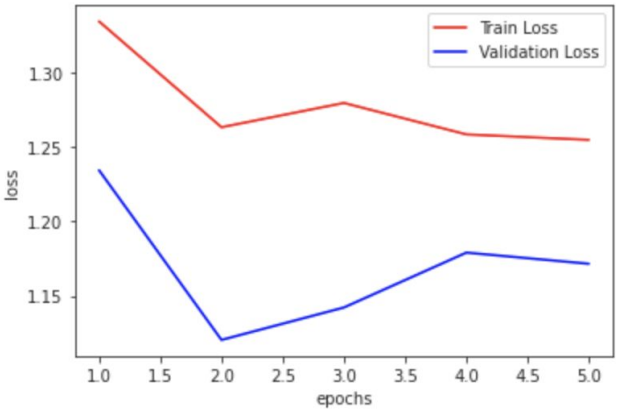
Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.967	nan	0.00	nan	0.756
Validation	0.970	nan	0.00	nan	0.759
Test	0.969	nan	0.00	nan	0.755

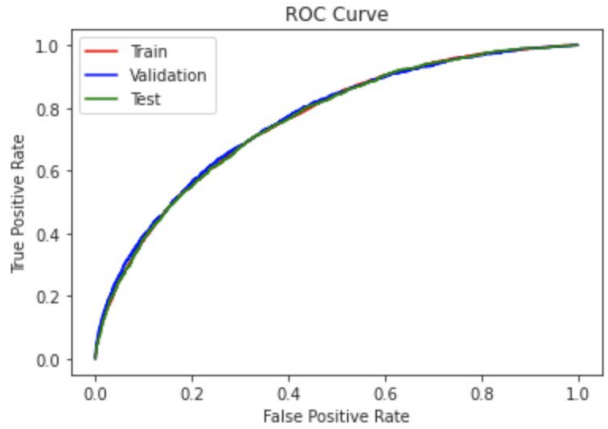




Weighted Binary Cross-Entropy Loss (epochs = 5)

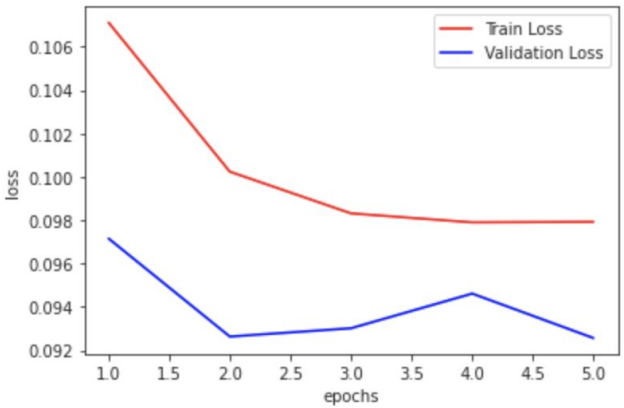
Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.826	0.09	0.50	0.16	0.759
Validation	0.828	0.09	0.51	0.15	0.762
Test	0.826	0.09	0.50	0.15	0.758

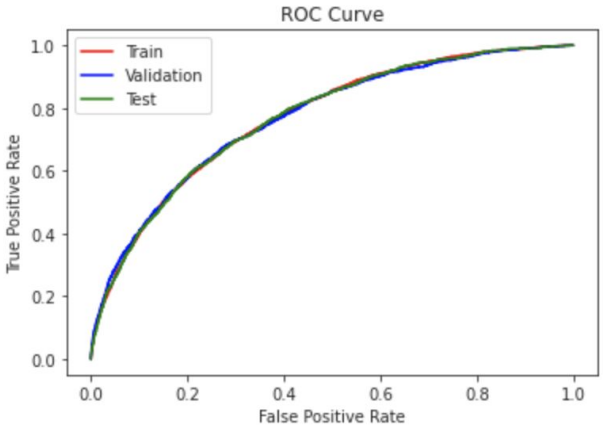




Focal Loss (epochs = 5)

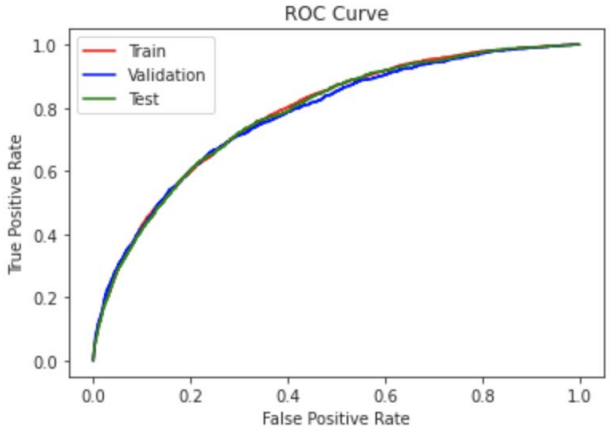
Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.954	0.20	0.14	0.17	0.769
Validation	0.957	0.21	0.15	0.17	0.769
Test	0.957	0.19	0.13	0.16	0.769





Focal Loss (epochs = 15)

Dataset	Accuracy	Precision	Recall	F1	AUC
Train	0.945	0.19	0.20	0.19	0.783
Validation	0.948	0.19	0.23	0.21	0.777
Test	0.947	0.18	0.20	0.19	0.780



Hierarchical Attention Network

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 100)	0	
embedding_1 (Embedding)	(None, 100, 100)	2000100	input_1[0][0]
bidirectional_1 (Bidirectional)	(None, 100, 100)	45300	embedding_1[0][0]
timedistributed_1 (TimeDistribut	(None, 100, 100)	10100	bidirectional_1[0][0]
attlayer_1 (AttLayer)	(None, 100)	100	timedistributed_1[0][0]

Total params: 2,055,600
Trainable params: 55,500
Non-trainable params: 2,000,100

The HAN was built as a functional model in keras.

The inputs are first passed at the sentence level to identify word attention embeddings and then the same model was processed with review data to identify sentence attention.

Based on a preliminary analysis, this model produced accuracy of over 93.25% in the first couple of epochs. Its full scope is still being explored on different loss functions.

Future Scope

- User/Item bias
- Role of context in sentences