Real or Not? NLP with Disaster Tweets

By Chris Reimann, Kyla Ronellenfitsch,
Devanshi Verma, Sai Prasanna Teja Reddy Bogireddy

Machine Learning & Predictive Analytics
Professor Arnab Bose

December 11, 2020

Agenda

1 Problem Statement

2 Data Properties & Transformations

3 Assumptions & Hypotheses

4 Proposed Approaches

5 Proposed Solution

6 Results

7 Challenges & Future Work

Problem Statement

Twitter's real-time, crowdsourced news has become an important communication channel in times of emergencies. Monitoring crisis-related tweets allows first responders, FEMA, and other disaster relief organizations to respond timelier and more effectively to disasters.

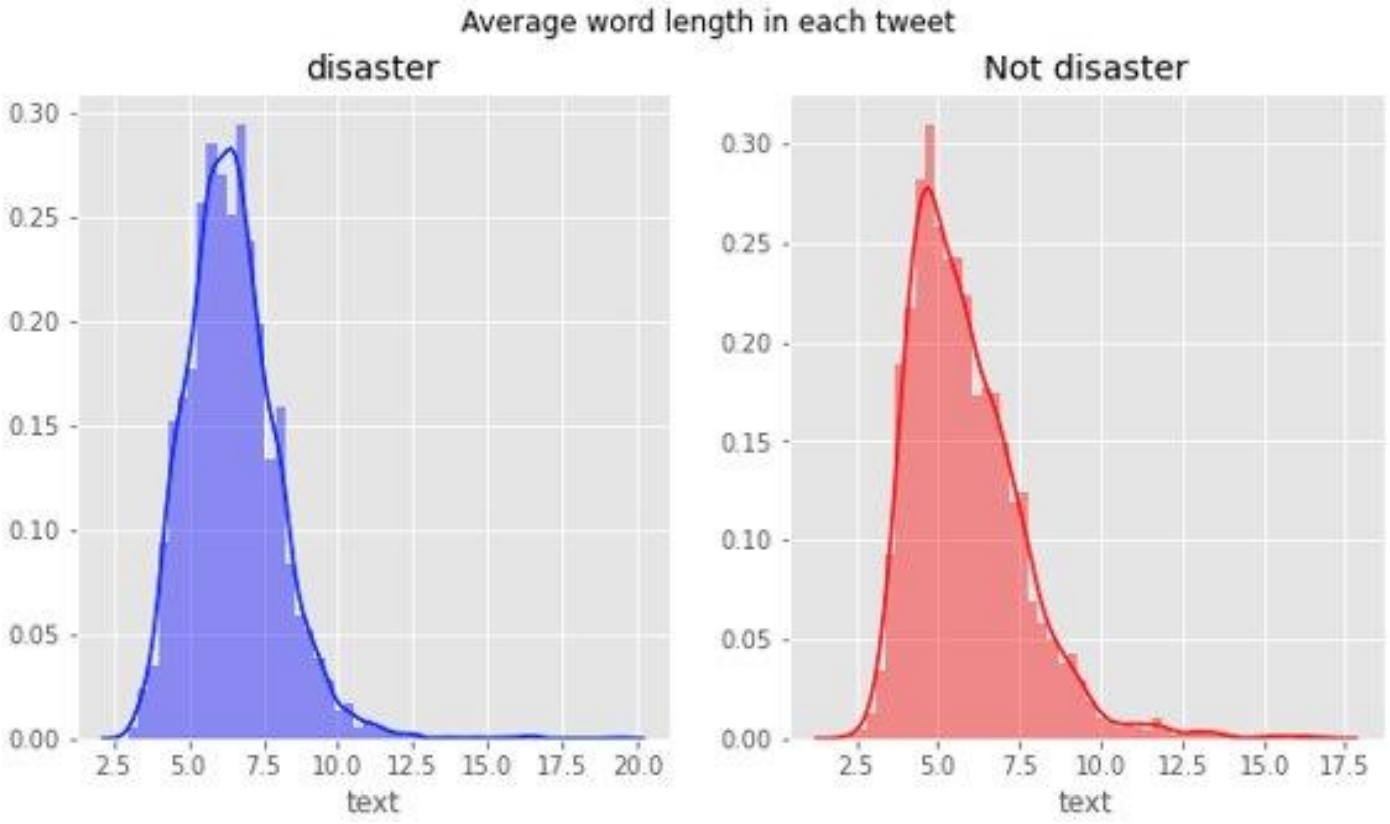
However, Twitter is only as useful as its information's validity. Slang, sarcasm and hyperbole make identifying true emergencies a challenge.

Thus, determining actual crisis-related tweets from non-crisis related tweets is a requirement in making Twitter useful in detecting actual crises.

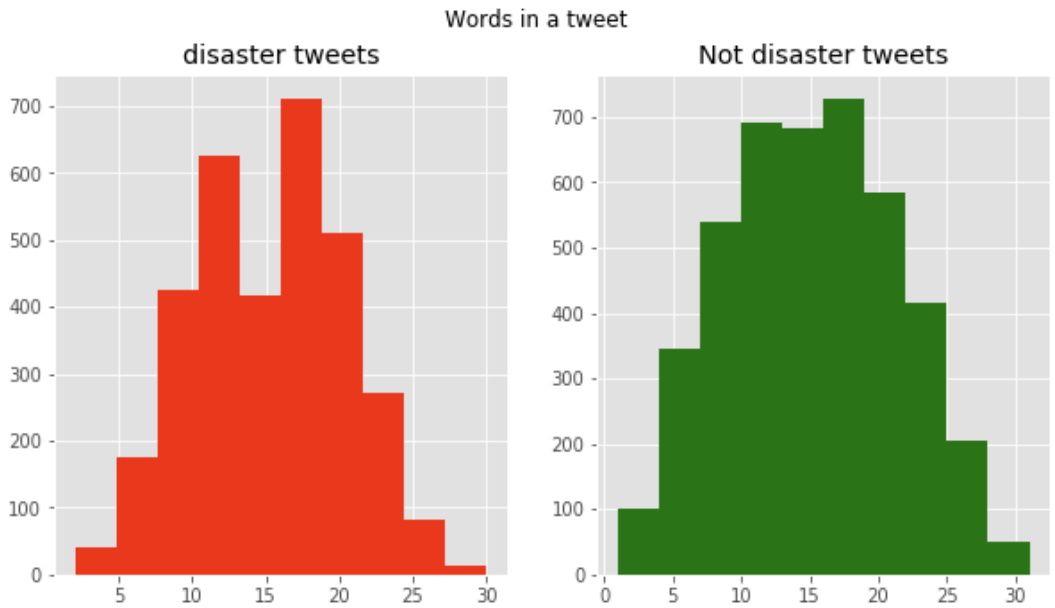
Data Properties

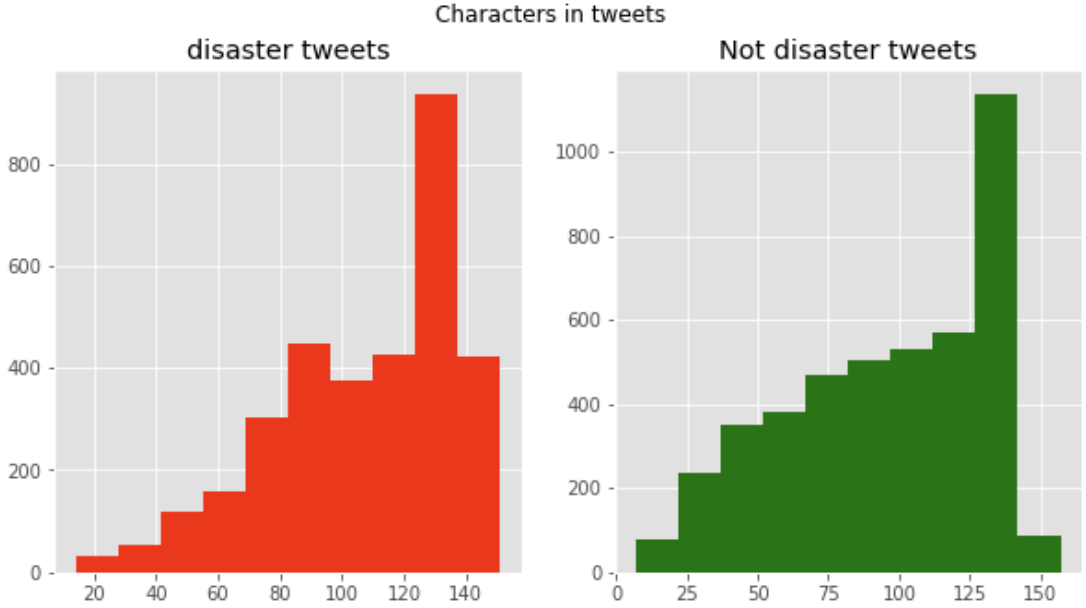
ID	Keyword	Location	Text	Target
1	Wreck	USA	"Don't think I Can take anymore emotional wreck watching @emmerdale #SummerFate @MikeParrActor @MissCharleyWebb",	0
2	Ablaze	Brighton, UK	'Deputies: Man shot before Brighton home set ablaze http://t.co/gWNRhMSO8k',	1
3	Tsunami	NAN	'#sing #tsunami Beginners #computer tutorial.: http://t.co/ukQYbhxMQI Everyone Wants To Learn To Build A Pc. Re http://t.co/iDWS2ZgYsa',	0

Data Properties



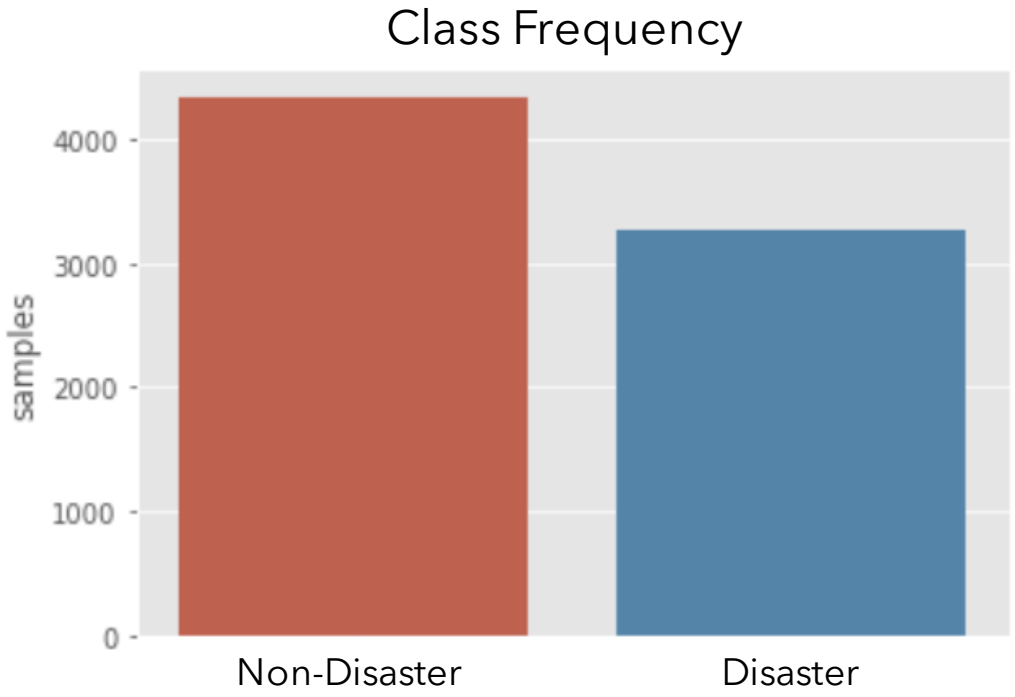
Data Properties





Data Properties

- Relatively balanced classes (40% Disaster | 60% Non-Disaster)

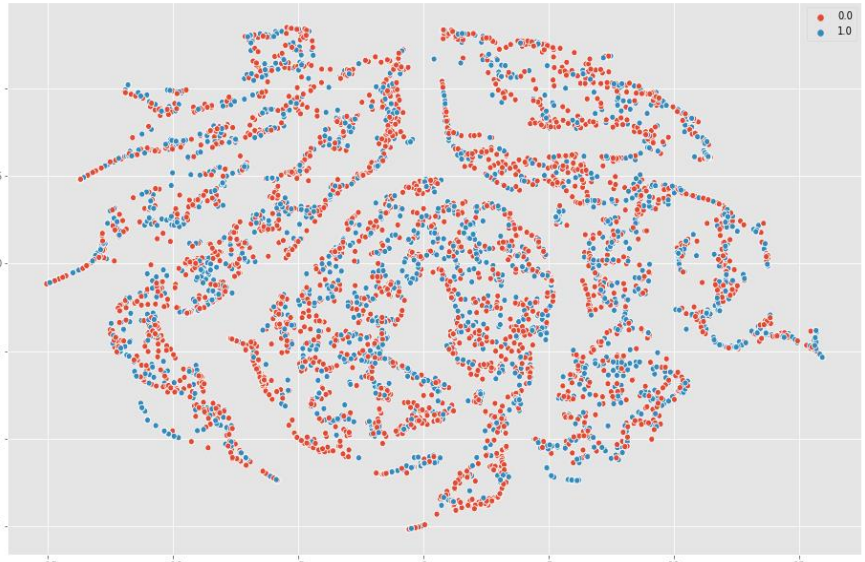


Data Transformations

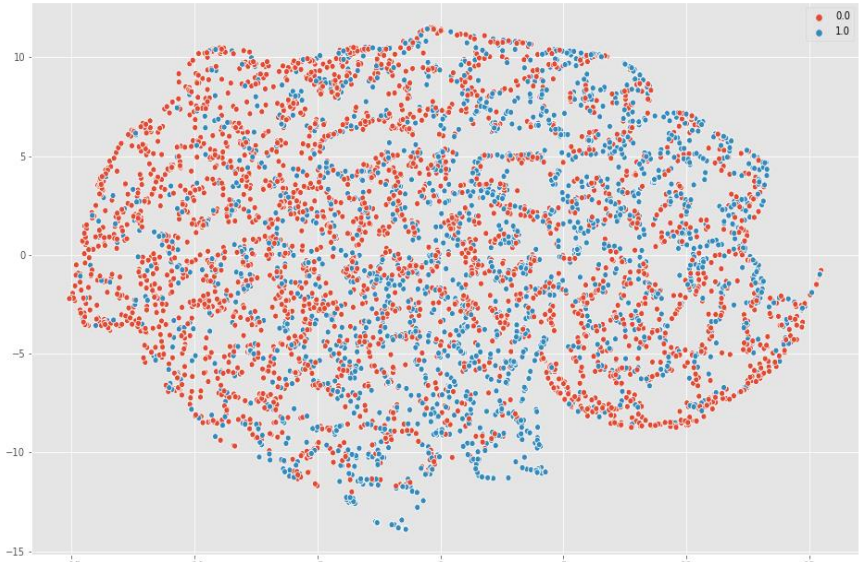
- Removal of
  - Emojis
  - Punctuation
  - URLs
  - HTML
- Stemming
- Convert to root word
- Convert to lowercase
- Correct spelling

Data Transformations

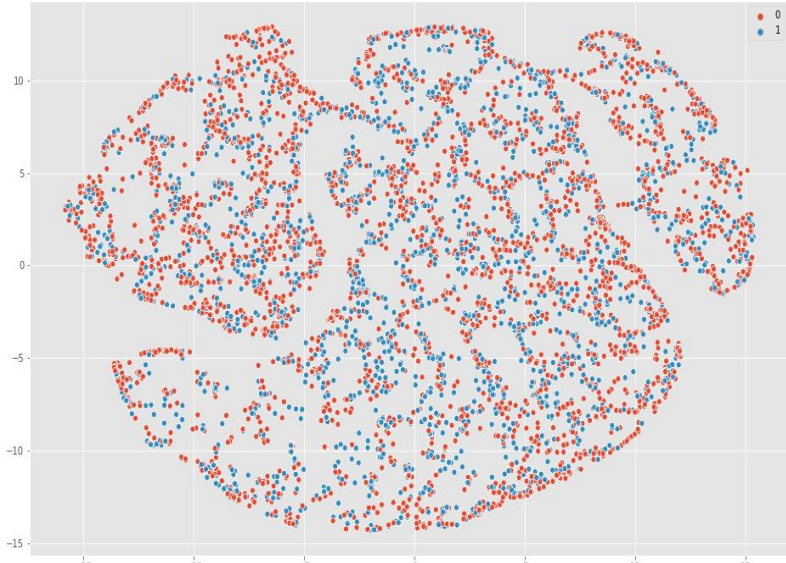
TSNE Plot using Count Vectorizer



TSNE Plot using TFIDF



TSNE Plot using GloVe



Assumptions & Hypotheses

- Deployment of emergency personnel is high stakes, but also resource-intensive. Therefore, we equally prioritize successful detection of both classes.
- Use F1 Score, which balances Precision and Recall as validation metric.

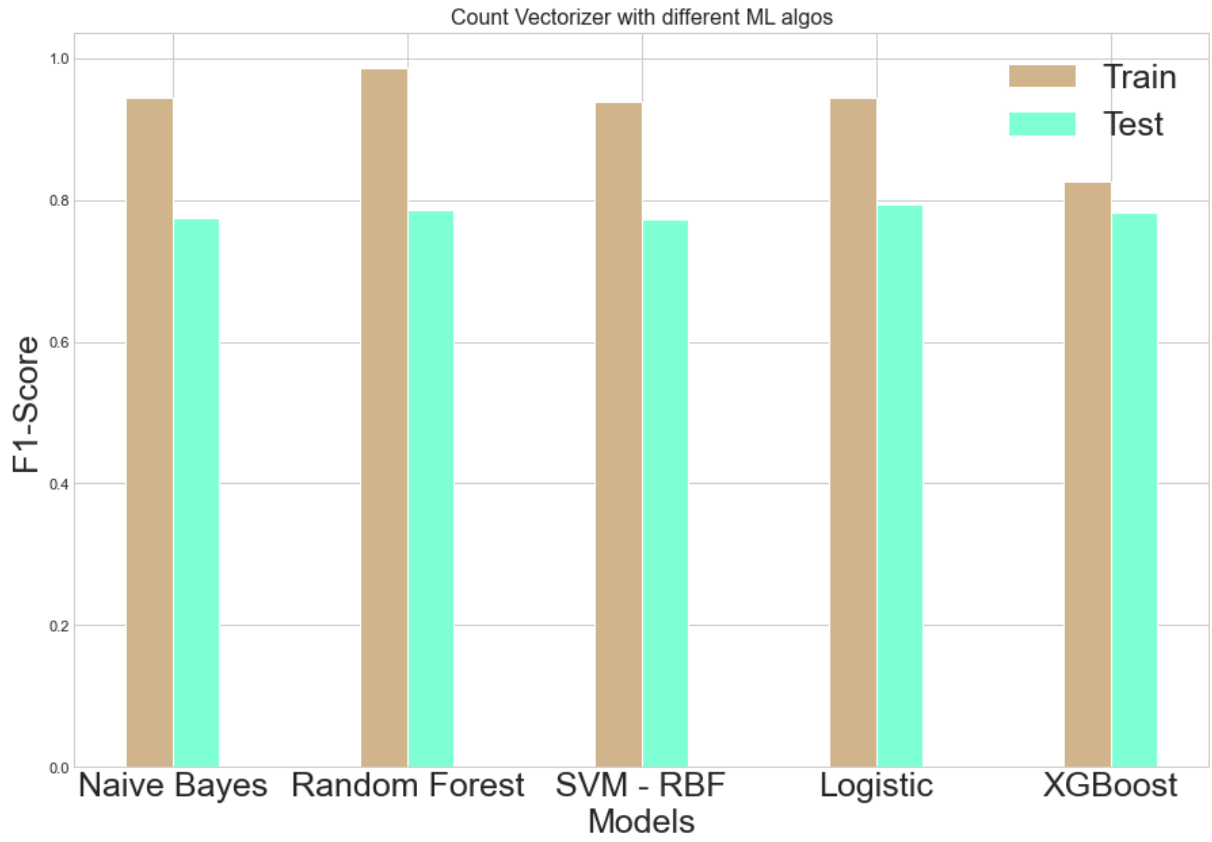
$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Proposed Approaches

- Classic Machine Learning Classifiers
  - Logistic Regression
  - Random Forest
  - SVM (Linear, RGB and Polynomial)
  - Gradient Boosting
  - AdaBoost
  - XG Boost
- LSTM
- Bidirectional LSTM
- Temporal Convolutional Network
- BERT

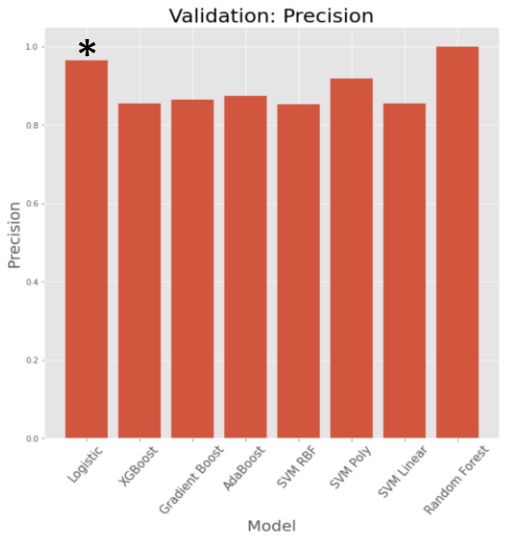
Classic Machine Learning Classifiers

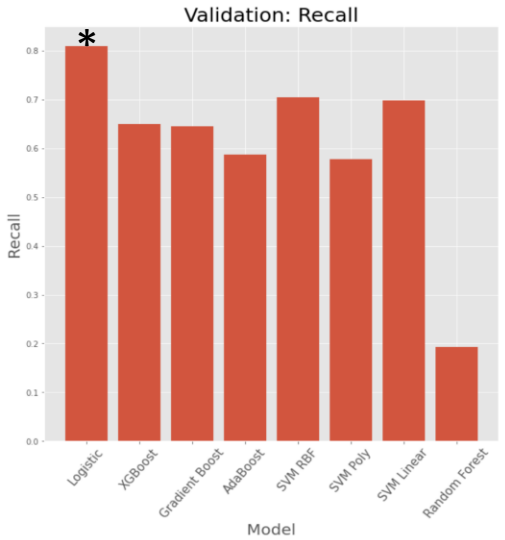
- Count Vectorizer using ‘text’ column
- Hyper-parameter tuning
- For generalizability we go for XGBoost but in terms of best f1 we go with Logistic

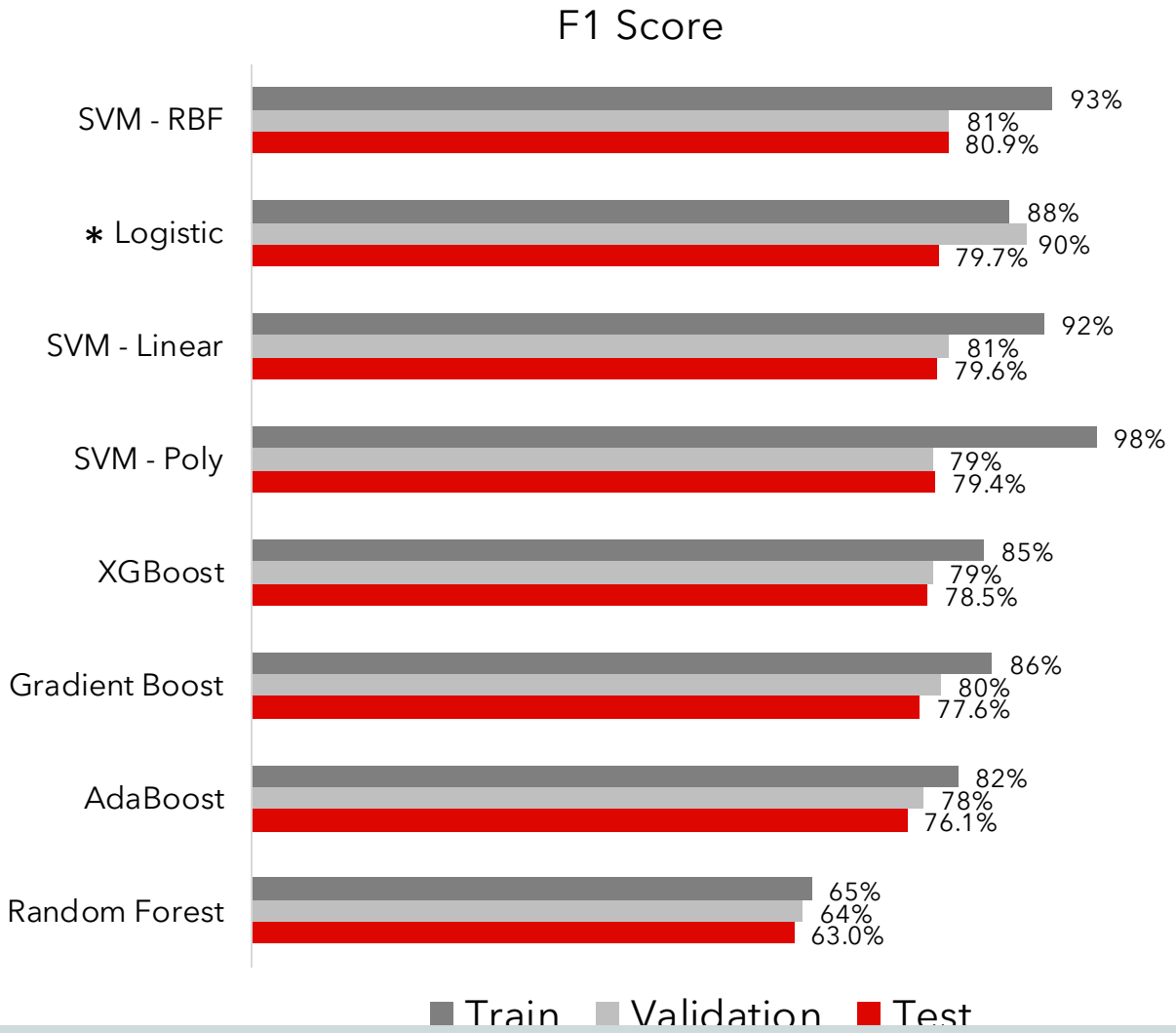


Classic Machine Learning Classifiers

- TF-IDF Vectorizer using 'text' column
- Hyper-parameter tuning
- SVM models overfit most
- Logistic Regression best model due to high precision and recall, high test F1 score, and lower degree of overfitting







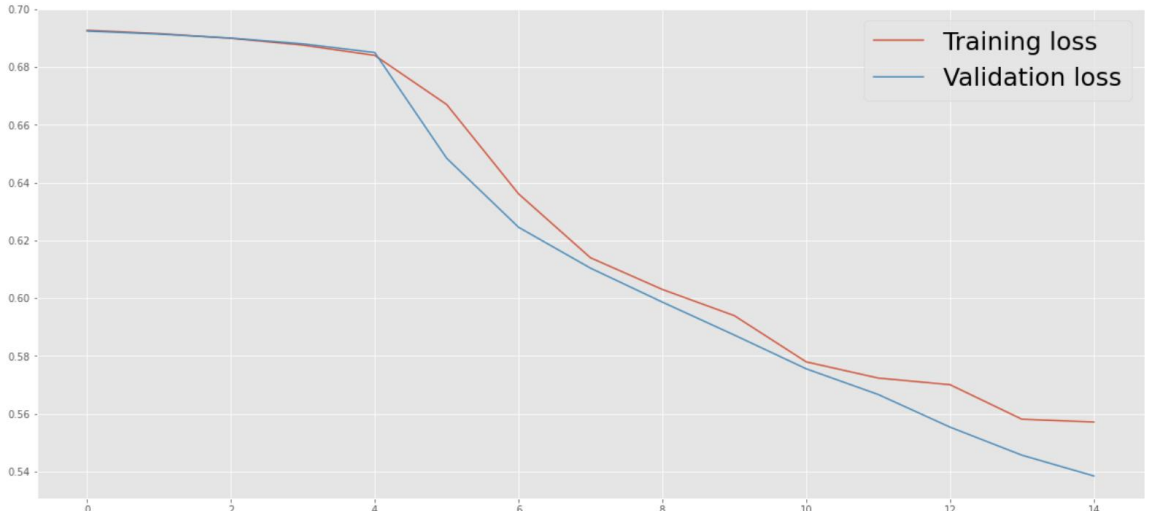
LSTM

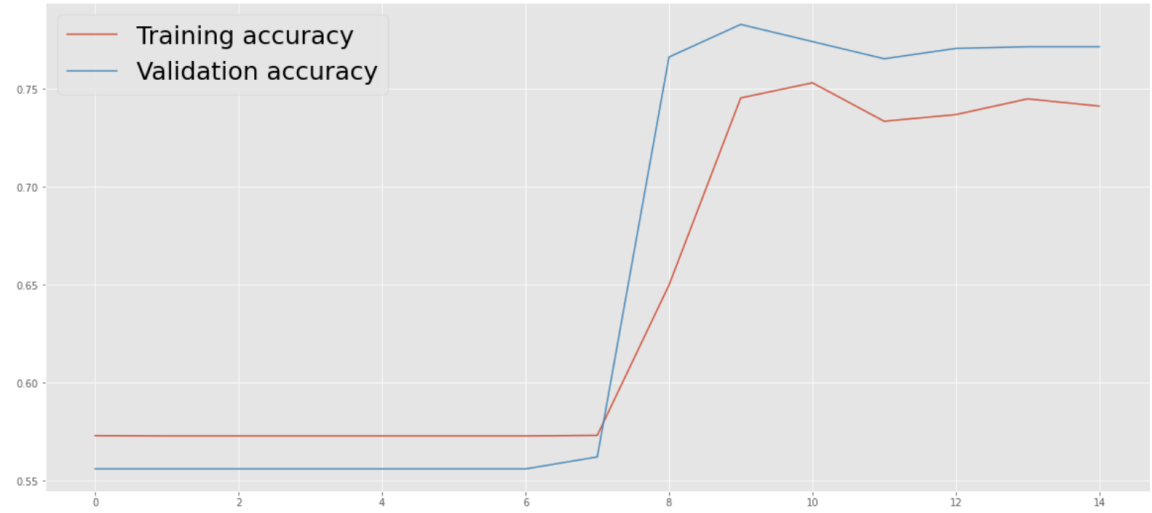
Architecture
- GloVe Embeddings
- Spatial Dropout
- LSTM
- Dense

Fit
- Adam Optimizer
- Batch Size = 38
- Epochs = 15

Train Precision = 76%
Train Recall = 76%

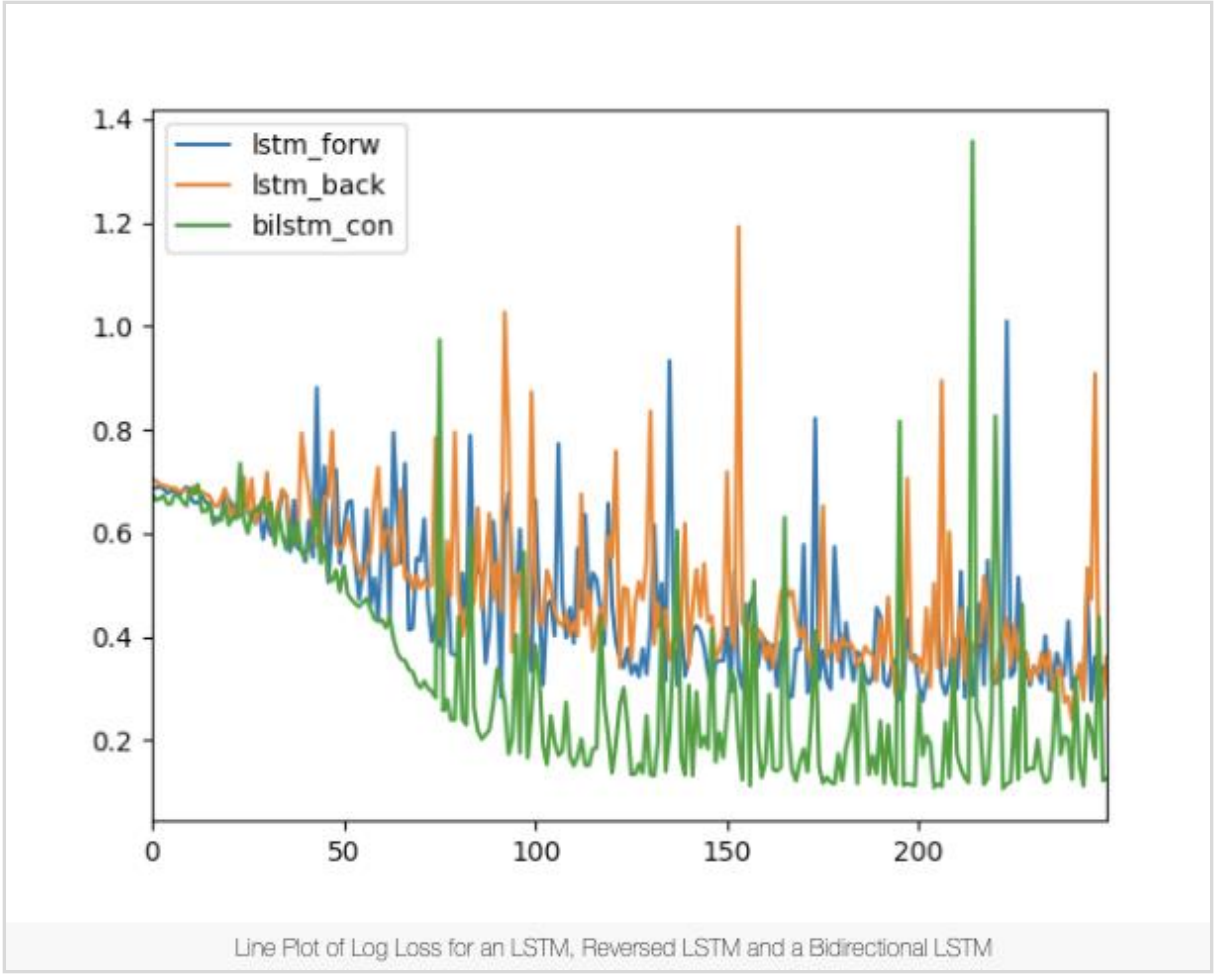
Kaggle Test F1 = 78.6%





Bidirectional LSTM

Bidirectional LSTMs are an extension of regular LSTMs. They can be used on sequences where the entire sequence on the input data is known (such as a sentence or a tweet). Bidirectional LSTMs train two LSTMs instead of one. One LSTM is trained on the normal input and another is trained on the reverse copy of the input.



Line Plot of Log Loss for an LSTM, Reversed LSTM and a Bidirectional LSTM

Bidirectional LSTM

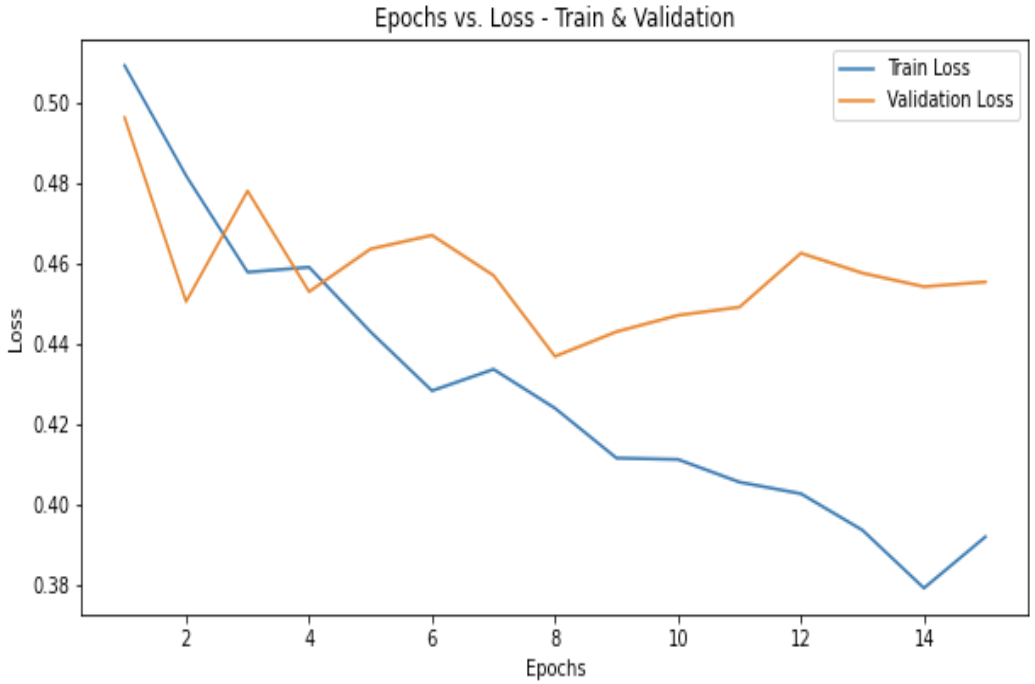
Why is a Bidirectional LSTM helpful for NLP?

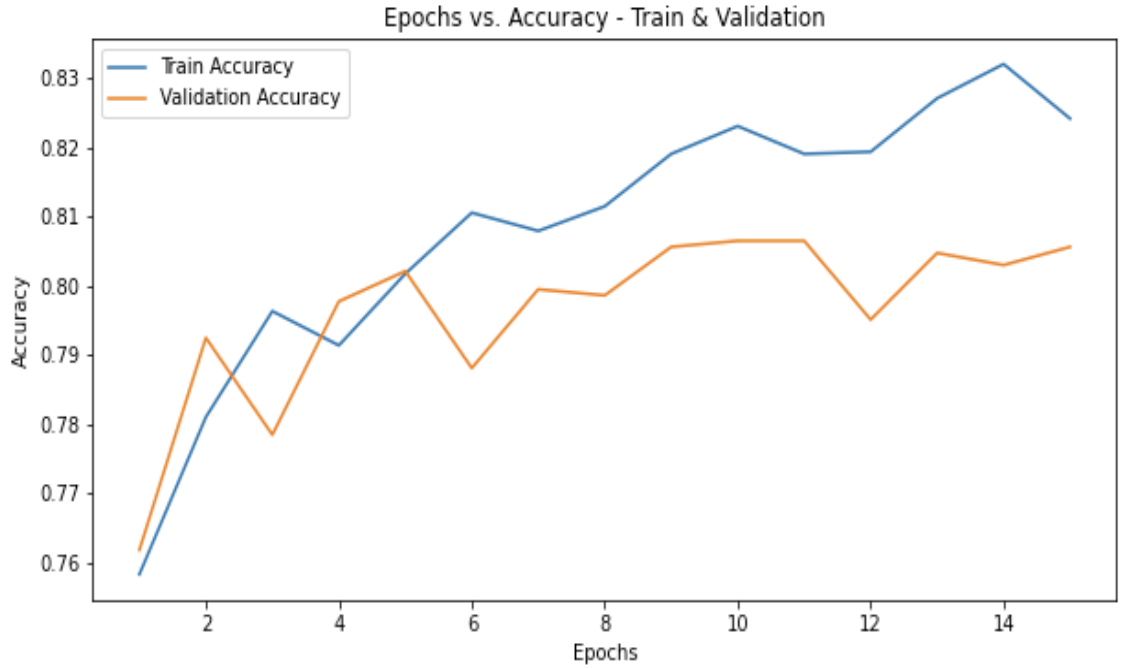
Complete ideas require future words to generate context.

> "… relying on knowledge of the future seems at first sight to violate causality. How can we base our understanding of what we've heard on something that hasn't been said yet? However, human listeners do exactly that. Sounds, words, and even whole sentences that at first mean nothing are found to make sense in the light of future context. What we must remember is the distinction between tasks that are truly online – requiring an output after every input – and those where outputs are only needed at the end of some input segment."

- Alex Graves and Jurgen Schmidhuber, Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005

Bidirectional LSTM

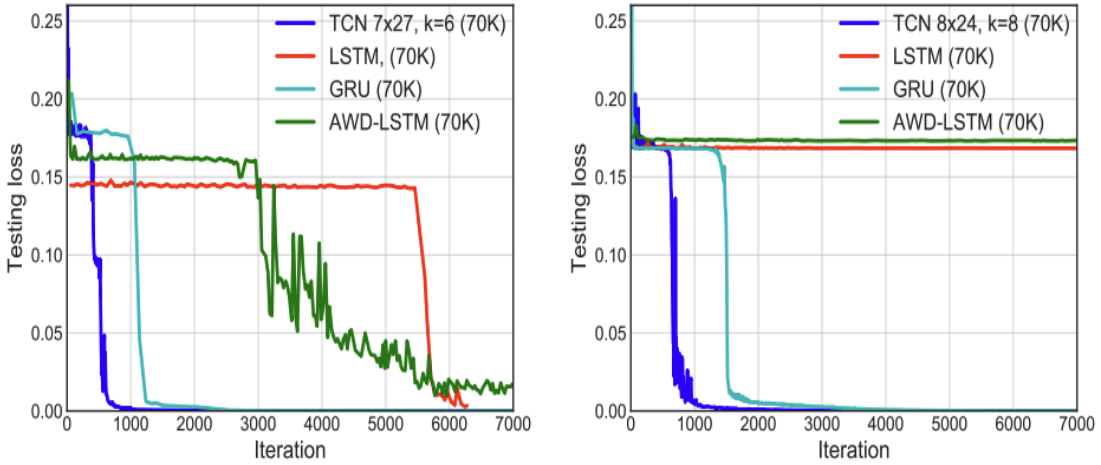




Kaggle F1 Test Score: 0.80907

Temporal Convolutional Network

For most deep learning practitioners, sequence modelling is synonymous with recurrent networks. Yet recent results indicate that convolutional architectures can outperform recurrent networks on tasks such as language modelling and machine translation[1].



(a) $T = 200$ (b) $T = 600$

Figure 2. Results on the adding problem for different sequence lengths T. TCNs outperform recurrent architectures.

[1] An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modelling, Shaojie Bai, J. Zico Kolter, Vladlen Koltun, 2018.

Temporal Convolutional Network

Even with the empirical results suggesting convolutional networks are more suitable than recurrent networks for sequence modelling tasks, they are still lost on us. This is a result of onset of attention-based models for language modelling. Hence, we tried to use this concept for language classification task.

Temporal Convolutional Network

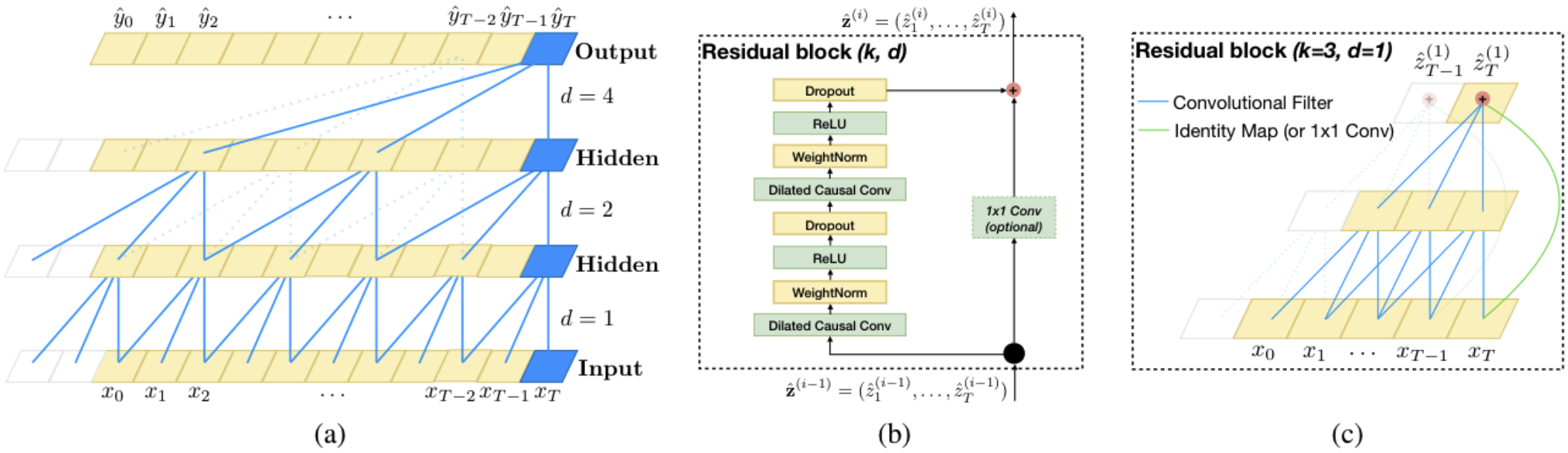
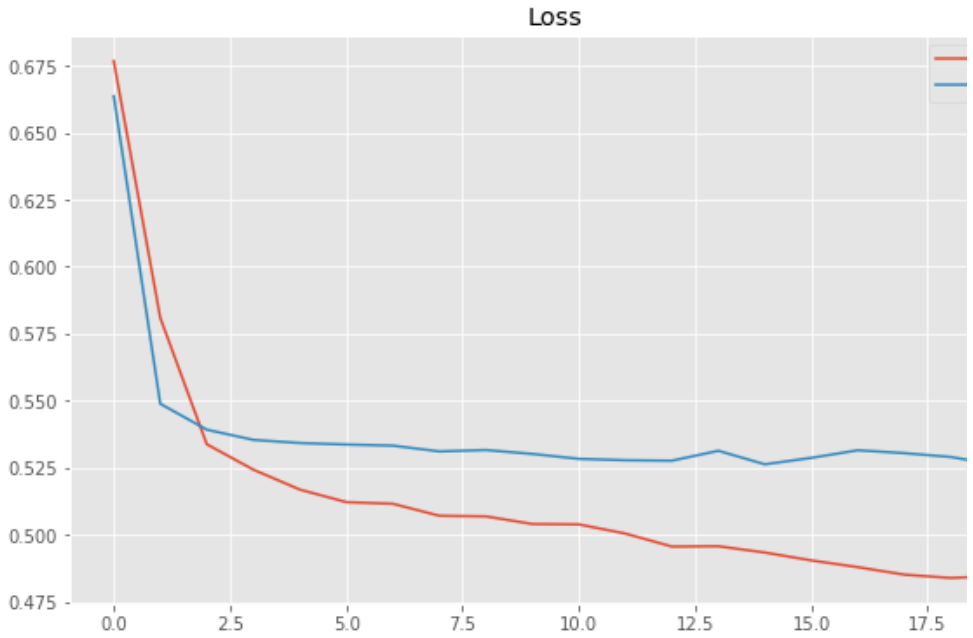
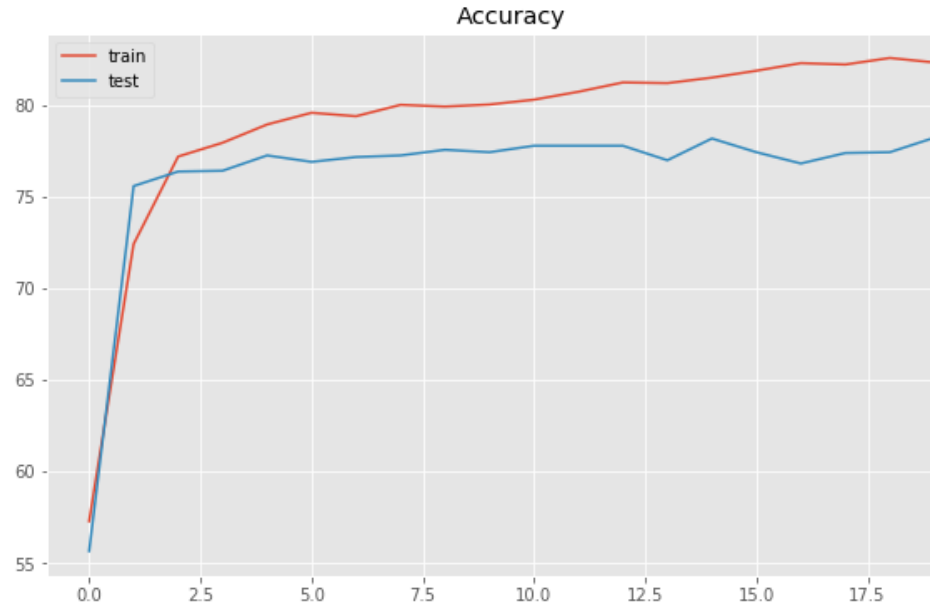


Figure 1. Architectural elements in a TCN. (a) A dilated causal convolution with dilation factors $d = 1, 2, 4$ and filter size $k = 3$. The receptive field is able to cover all values from the input sequence. (b) TCN residual block. An 1x1 convolution is added when residual input and output have different dimensions. (c) An example of residual connection in a TCN. The blue lines are filters in the residual function, and the green lines are identity mappings.

Temporal Convolutional Network

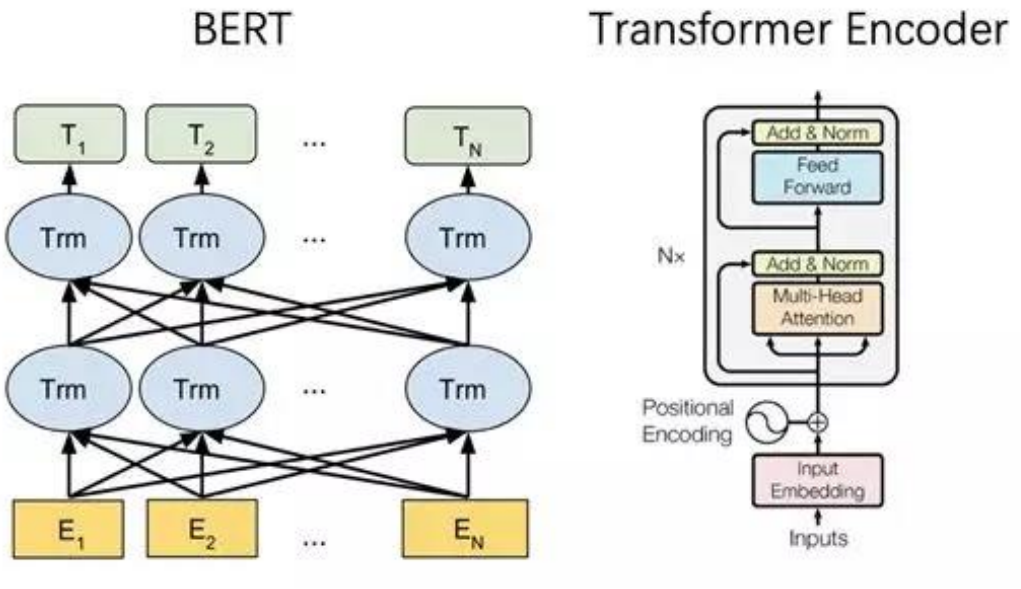




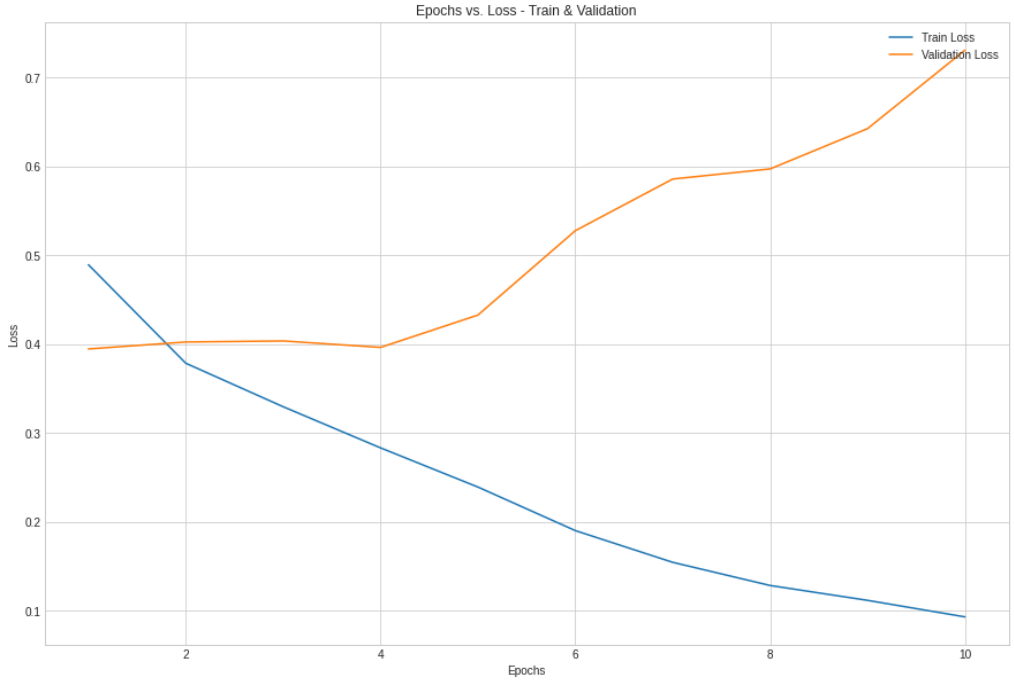
Kaggle F1 Test Score: 0.81244

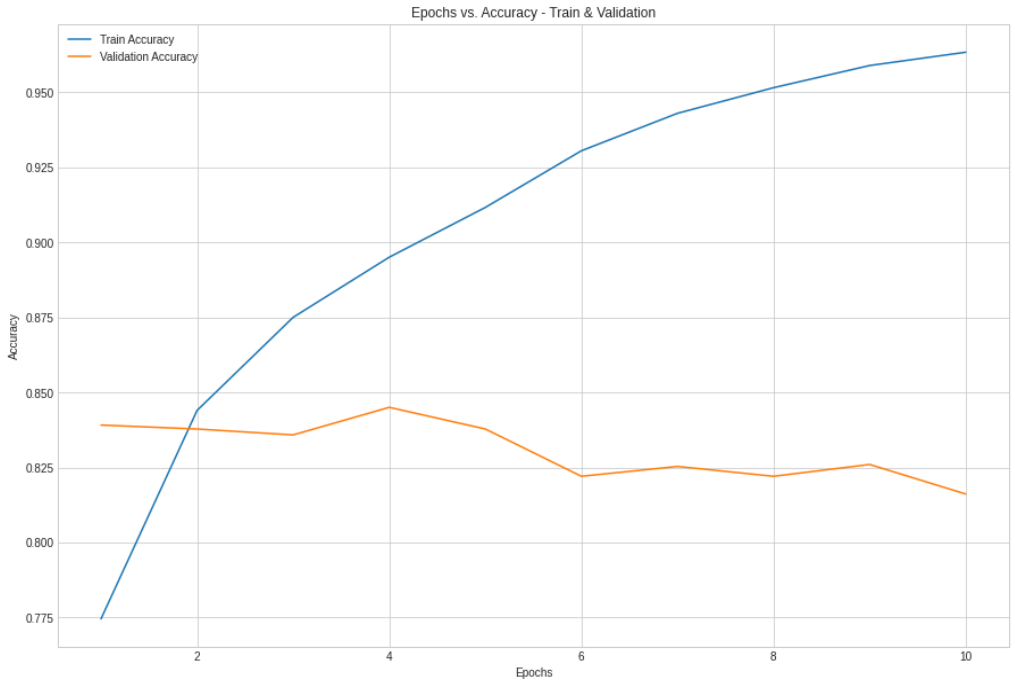
Bert

- **Self Attention:** Transformer based models allow self-attention which means no locality bias
- **Efficiency:** Single multiplication per layer which means efficiency over TPU's
- **Masking:** Mask out k% of the input words, and then predict the masked words
- **Embeddings:** Sum of three embeddings



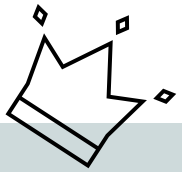
Bert





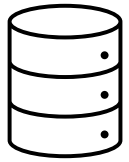
Kaggle F1 Test Score: 0.84247

Model Comparison



Architecture	CV Logistic	TFIDF Logistic	LSTM	Bi-Directional LSTM	Temporal CNN	BERT
Kaggle Score (F1)	0.793	0.79681	0.78639	0.80907	0.81244	0.84247
Pros	Simple; accurate	Simplicity	Remembers sequence/text data over long gaps	Future values context	Can capture longer dependencies and faster convergence	Contextual; efficient
Cons	Worse than state of the art algos	Assumptions of linearity, captures less complex relationships	Computationally expensive	Computationally costly vs. LSTM	Perform worse than Transformers	Interpretability

Challenges and Future Work



- Kaggle-curated dataset
- Actual tweets are probably not as balanced between actual disasters and not disaster
- Scraping our dataset for one geographical

- Similar results despite complexity of the models

- Incorrect labelling

Team

Chris Reimann

Kyla Ronellenfitsch

Teja Reddy

Devanshi Verma

THE UNIVERSITY OF CHICAGO

References

- Github: https://bitly.com/2W3nPub

Questions