Fake News Detection Using Deep Learning

ECE 884 - Deep Learning

20 April, 2021

Namratha Shah, Sneha Mhaske, Nat Hawkins

Group 4

Overview

Major boom in the field starting in 2015 (Oshikawa et al., 2020)

Determining the legitimacy of information source

Largely motivated by political news sources, important for proliferating new scientific information (Girgis et al., 2018)

Available articles online increases exponentially every day, too much for humans to inspect and filter by hand

Automated solutions for processing large volumes of information are required

Existing Approaches

Variety of approaches have been used to address this problem (as well as combinations thereof)

- **Fully Connected DNN** (Thota et al., 2018; Saikh et al., 2020)
- **LSTM** (Rodriguez and Iglesias, 2019; Kumar et al., 2019)
- ***Bi-LSTM** (Popit et al., 2018; Qawasmeh et al., 2019; Abedalla et al., 2019; Kumar et al., 2019)
- **CNN** (Rodriguez and Iglesias, 2019; Kumar et al., 2019)
- **Graph CNN** (Monti et al., 2019)
- **RNN** (GRU, "Vanilla") (Girgis et al., 2018; Singhania et al., 2017)
- ***BERT** (Abedalla et al., 2019; Zellers et al., 2020)
- **"Simple Models"** (e.g., Log Reg, SVM)

Variety of features: CBOW, TFIDF, N-grams, word2vec, fastText, GloVE, ELMo

Proposed Approach - LSTM

LSTM: Only remember the past inputs

BiLSTM: Remember both past and the future

Proposed Approach - LSTM

LSTM:

- Sequential model
- Embedding layer
- LSTM - 2 layers
- Dense - 2 layers

BiLSTM:

- Non-sequential(functional) model
- Embedding layer
- Bidirectional - 2 layers
- Dense layer

Proposed Approach - Transformer for Classification

Generative Pre-training GPT-2

GPT-2 is trained with a simple objective: predict the next word, given all of the previous words within some text.

A semi-supervised approach for language understanding tasks using a combination of unsupervised pre-training and supervised fine-tuning.

Employ a two-stage training procedure. First, use a language modeling objective on the unlabeled data to learn the initial parameters of a neural network model. Subsequently, adapt these parameters to a target task using the corresponding supervised objective.

Uses a multi-layer Transformer decoder for the language model.

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

. Architecture hyperparameters for the 4 model sizes.

Proposed Approach - Transformer for Classification

Pre-trained BERT - Bidirectional Encoder Representations from Transformers

Notes that the major limitation of standard language models is that they are unidirectional.

BERT uses Pre-trained Transformer Bi-directional Encoder which can then be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks without substantial architecture modifications.

Uses a “masked language model” (MLM) pre-training objective which enables the representation to fuse the left and the right context.

Also pre-trained on Next Sentence Prediction task which is helpful for tasks like Question Answering.

For fine-tuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks.

BERTBASE (L=12, H=768, A=12, Total Parameters=110M) and BERTLARGE (L=24, H=1024, A=16, Total Parameters=340M)

Baseline Models

- 9 different pre-trained embedding methods from flairNLP python library
  - Flair-news-forward (1024),
  - Flair-news-backward (1024),
  - Flair-multi-forward (2048),
  - Flair-multi-backward (2048),
  - ELMo (3072),
  - Twitter (100),
  - GloVe (100),
  - Fasttext (300),
  - Bert (1024)
- 4 different feature sets. Creating embeddings from
  - Title only
  - Body only
  - Concatenation of title and body
  - Average of title and body

Baseline Models

- 10 Sklearn models (default parameters):
    - L1 logistic regression,
    - L2 logistic regression,
    - KNN (k = 3)
    - KNN (k = 5)
    - KNN (k = 10)
    - Linear SVM
    - RBF SVM
    - Multilayer perceptron
    - Gaussian kernel NB
    - QDA

Dataset

[Fake News Dataset (Kaggle)]

Curated from existing works (Ahmed et al., 2017; Ahmed et al., 2018)

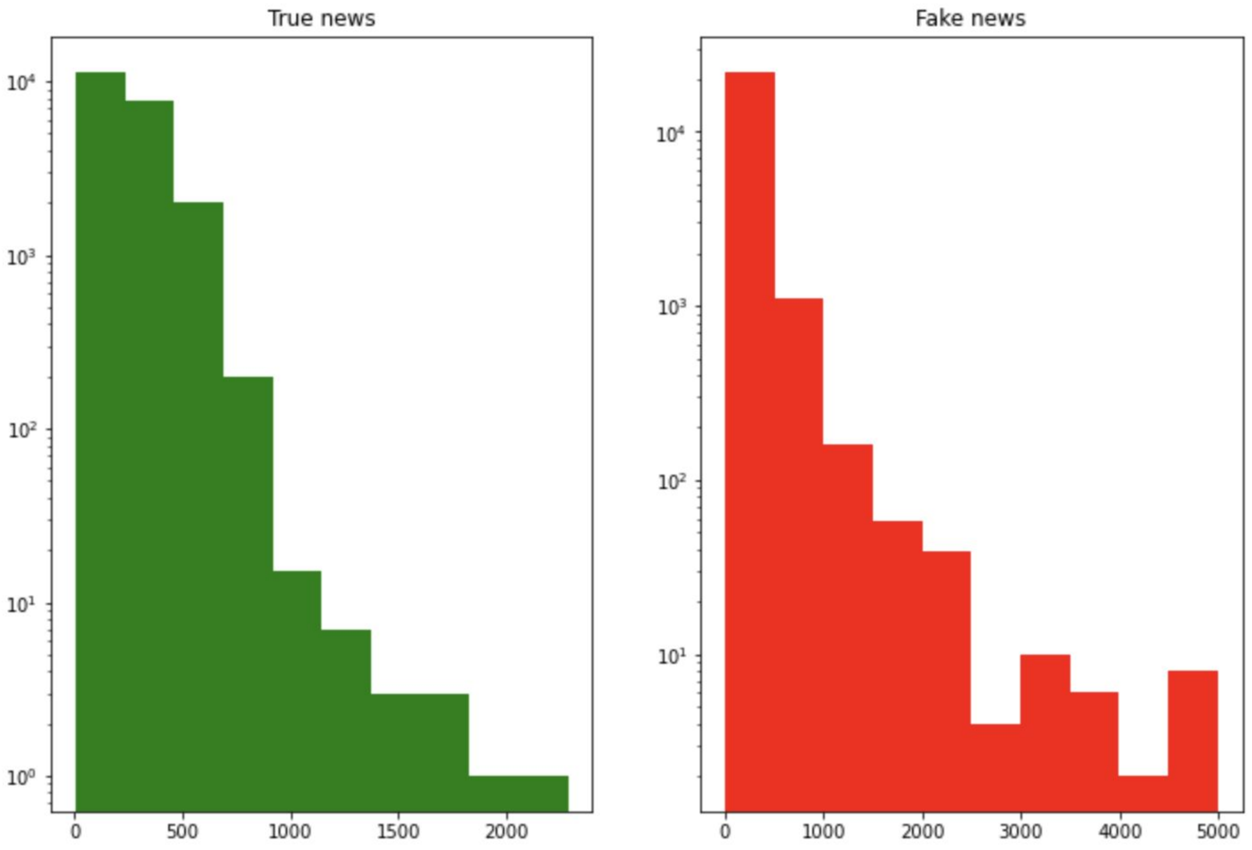
~20,000 titles and articles for each class from variety of years

Dataset

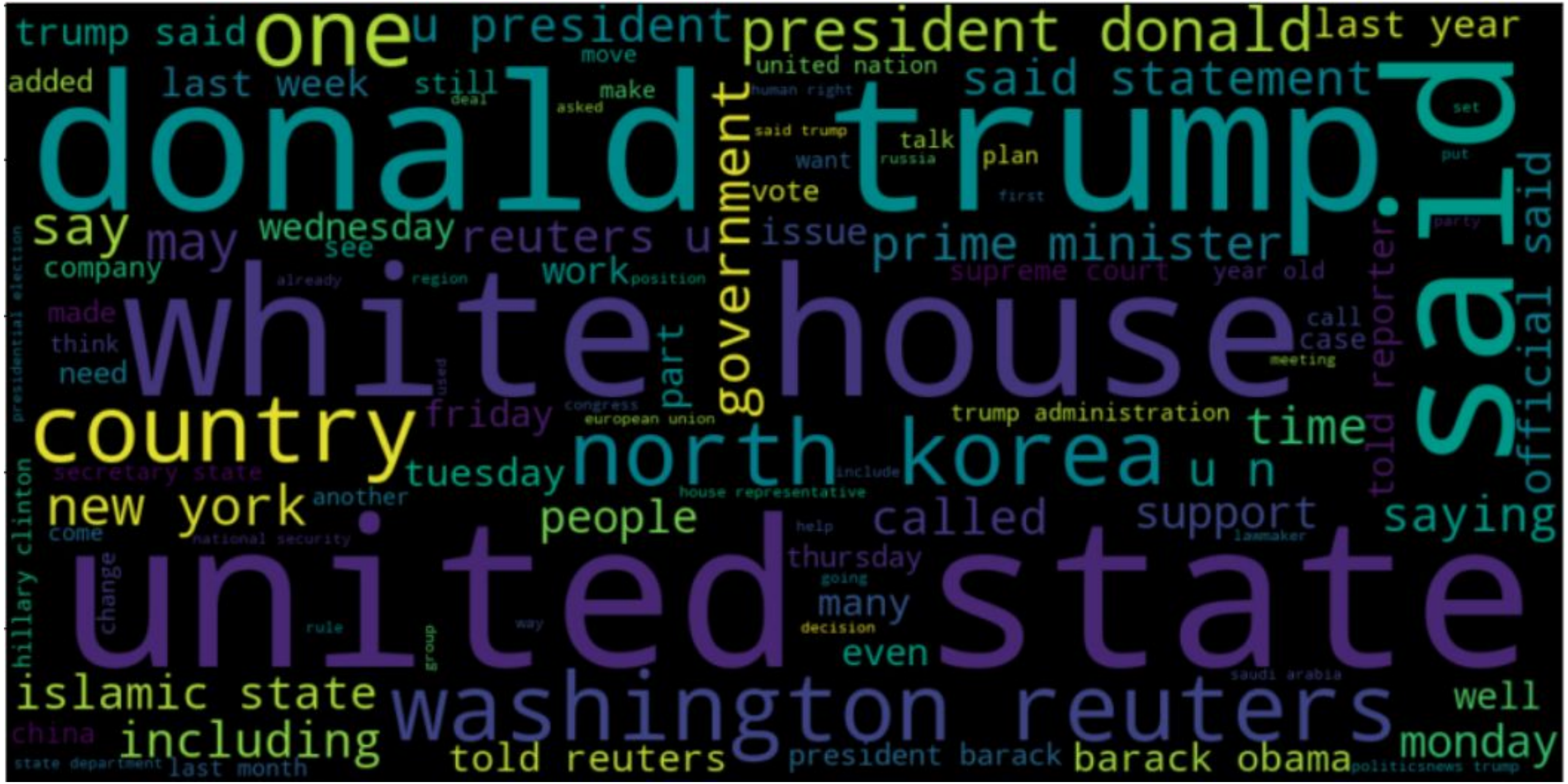
Subject	Count
Political News	11,272
World News	10,145
News	9,050
Politics	6,841
Left-News	4,459
Government News	1,570
US News	783
Middle-East	778

Type	Count
True	21,417
Fake	23,481

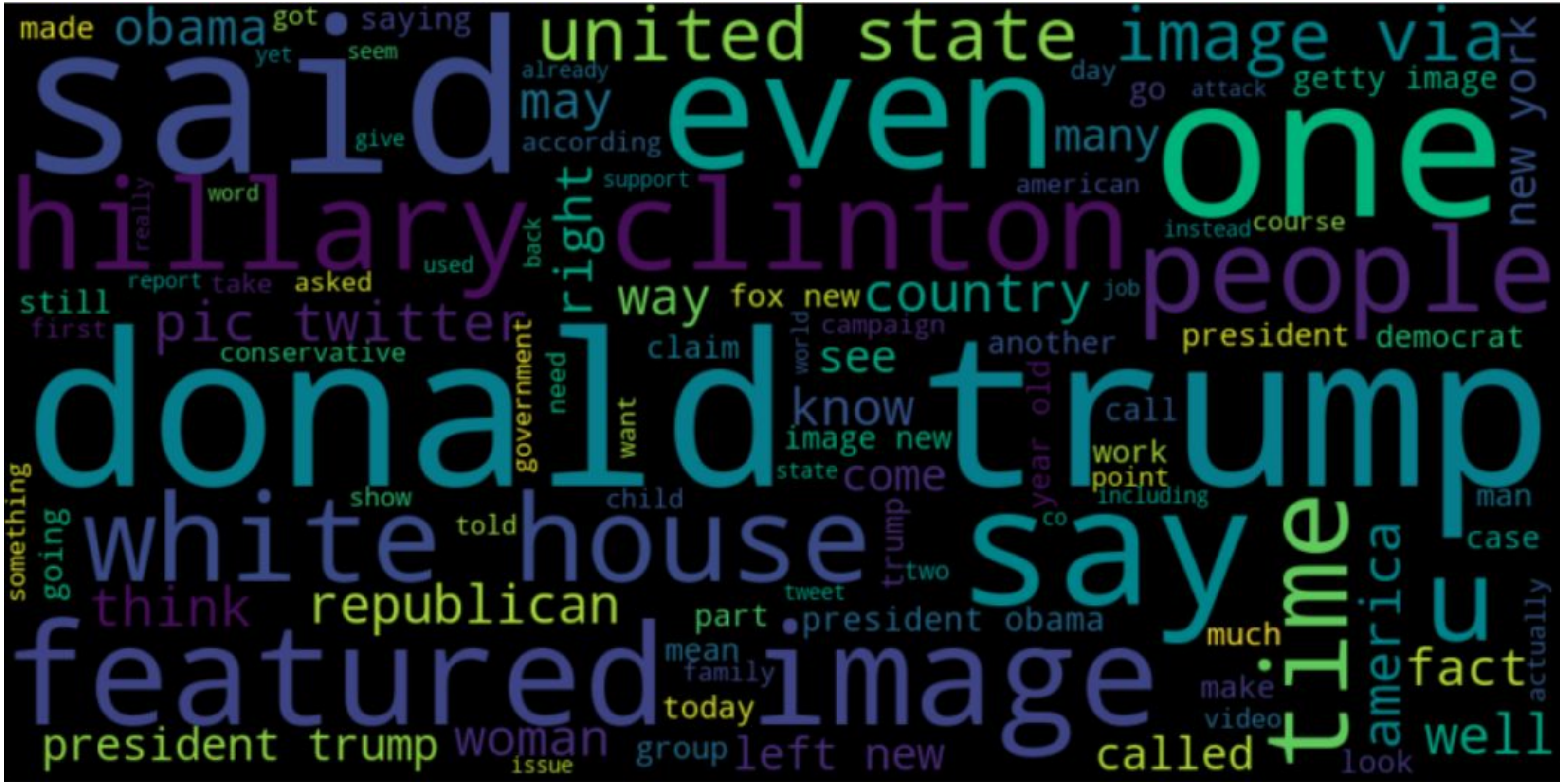
Dataset



Dataset


donald trump
white house
united state
said
washington reuters
north korea
country
one
president donald
u president
trump said
last year
said statement
government
prime minister
reuters u
last week
still
added
say
may
wednesday
new york
islamic state
including
told reuters
barack obama
president barack
monday
well
saying
support
called
people
tuesday
time
official said
told reporter
trump administration
friday
part
work
company
made
think
need
many
thursday
even
hillary clinton
secretary state
another
china
last month
state department
vote
issue
plan
talk
want
make
move
united nation
supreme court
year old
call
case
see
u n
come
change
politicsnews trump

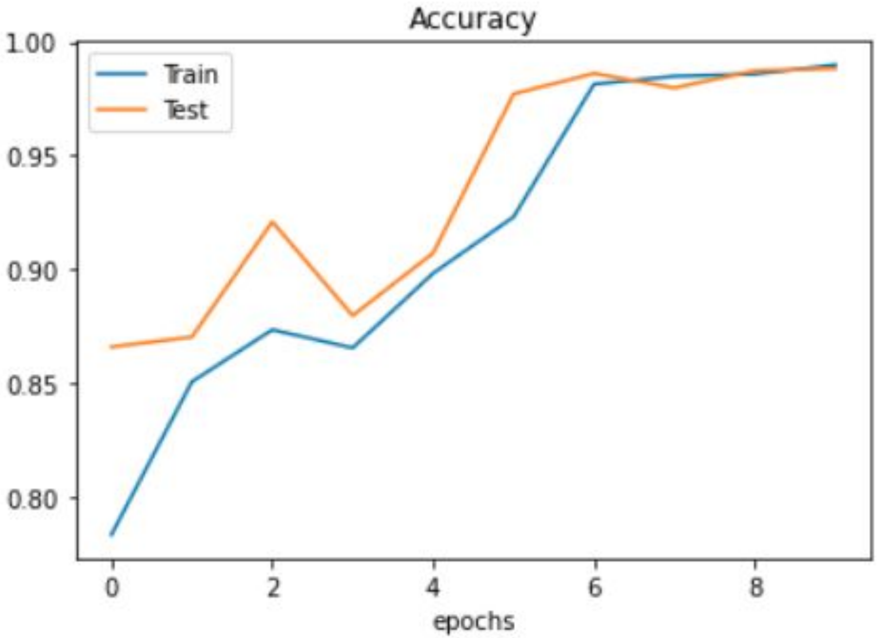
Dataset

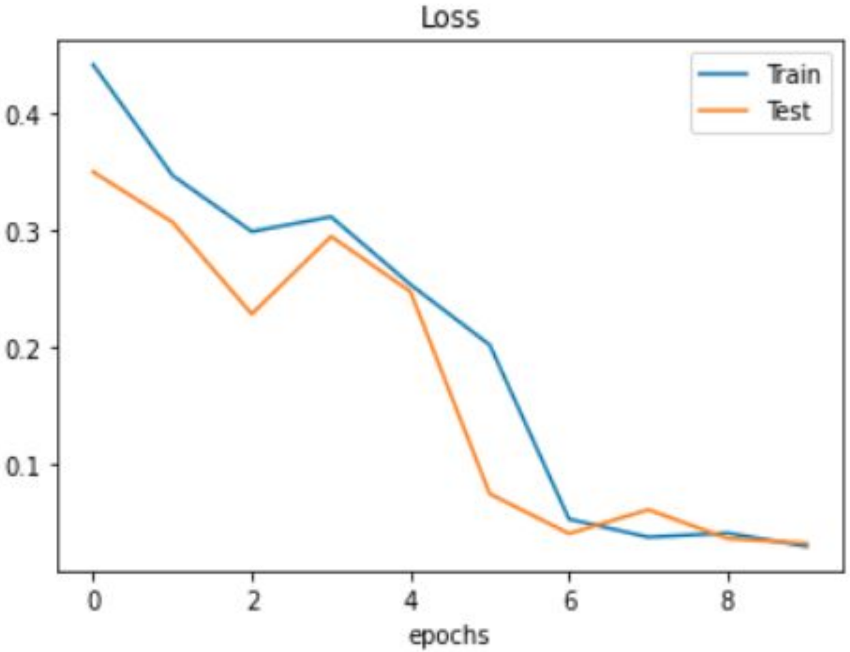

made
obama
got
saying
united state
image via
said
even
yet
seem
already
day
may
go
attack
getty image
new york
give
according
many
one
support
american
hillary
clinton
word
really
back
right
instead
course
people
used
report
take
asked
way
fox new
country
job
still
pic twitter
first
campaign
president
democrat
conservative
claim
world
another
see
donald
trump
know
call
need
government
want
year old
image new
work
state
come
point
something
going
show
child
including
man
told
white house
say
trump
co
time
case
america
two
think
republican
tweet
part
president obama
u
much
fact
featured
image
mean
family
actually
make
today
video
well
president trump
woman
group
left new
called
issue
look

Results Baseline Models

- Approximately 360* baseline models
- Top model for each feature set
  - Title only: MLP trained on elmo embeddings 95.7% accuracy (MLP bert: 95.1%)
  - Body only: MLP trained on bert embeddings 95.1% accuracy (L1 log reg elmo: 94.2%)
  - Concatenation: MLP trained on bert embeddings 95.1% accuracy (L1 log reg elmo: 94.3%)
  - Average: MLP trained on bert embeddings 95.1% accuracy (L1 log reg elmo: 94.3%)
- Summary:
  - Best performing baseline models were roughly 95% accuracy
  - Best embeddings: bert, elmo
  - Best models: MLP, L1 log reg, L2 log reg
  - “Good” models: SVM (RBF/Linear), QDA

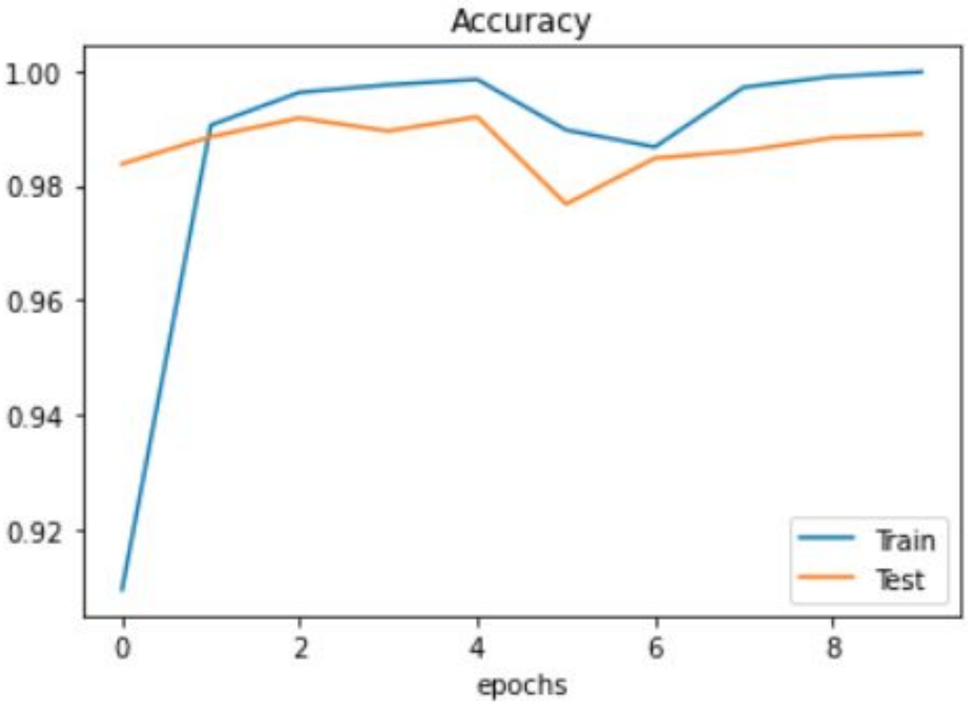
Results LSTM

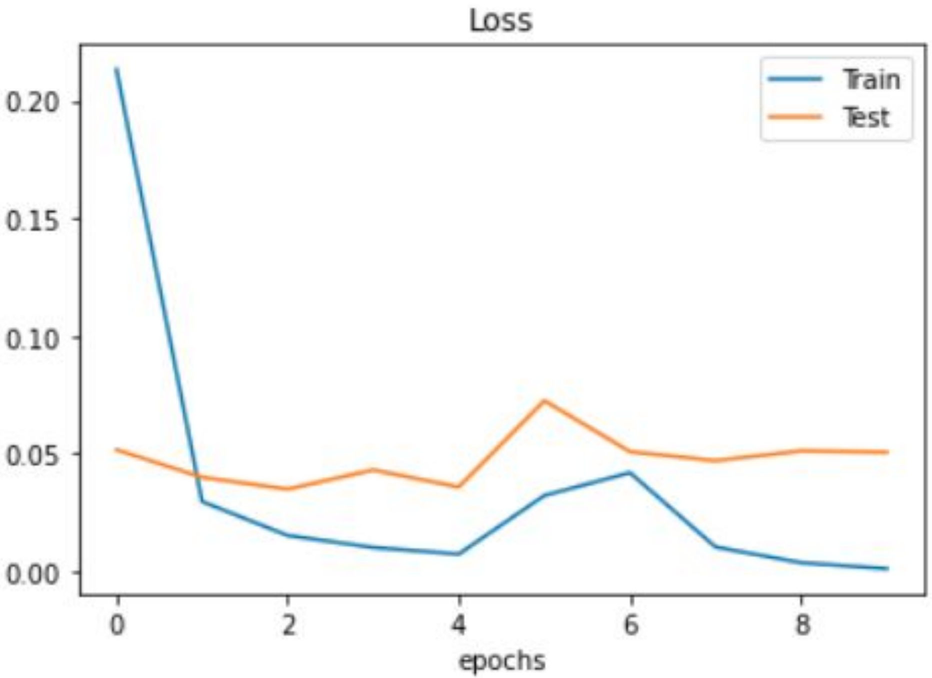




Final Test Set Accuracy: 98.99%

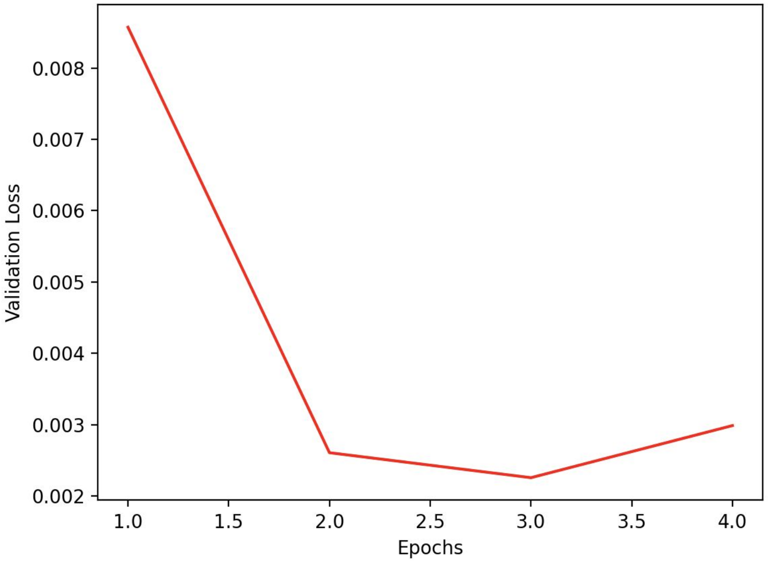
Results Bi-LSTM



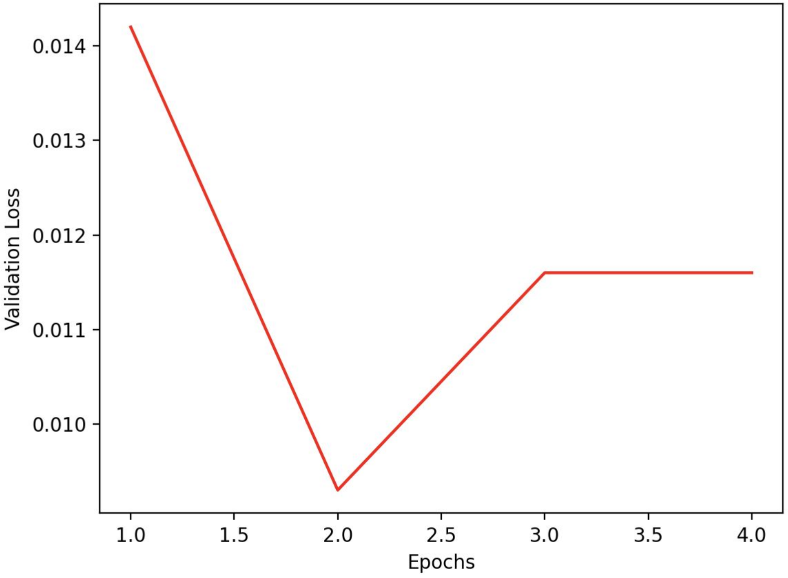


Final Test Set Accuracy: 98.83%

Results Transformer



BERT



GPT-2

Training Details for Transformer based models

Basic models for both BERT and GPT-2 were used - 12 layers with 100M+ parameters

For both GPT-2 and BERT, we simply add a classification linear layer to the pre-trained architectures and requires only 2-4 epochs.

Trained on Tesla v100 32G GPU with batch-size - 16, max_token_length - 512, learning rate - 2e-5, Adam Optimizer and learning rate decay schedule.

80-20 split between training and validation sets.

100% accuracy on this dataset from epoch 1 for both BERT and GPT-2.

Discussion, Conclusions, and Impact

The top performing baseline models are easy to train and perform comparably very well on this dataset (over 95% accuracy)

Current SOTA: >99% (task depending)

- Transformers are still being explored in this space (not enough to compare to)

Computationally lightweight solutions

Comprehensive results covering many different areas for this task

- Thorough benchmarking on this dataset for this task

Questions?

Extra Slides

Top 10 Baselines (Body)

Model	Accuracy
bert_MLP	0.951
elmo_L1 Log Reg	0.942
bert_QDA	0.941
elmo_L2 Log Reg	0.94
bert_RBF SVM	0.939
fasttext_MLP	0.933
bert_Linear SVM	0.929
flair-news-forward_MLP	0.929
bert_L1 Log Reg	0.928
bert_L2 Log Reg	0.928

Bottom 10 Baselines (Body)

Model	Accuracy
flair-multi-forward_Naive Bayes	0.77
flair-news-forward_Naive Bayes	0.769
flair-multi-backward_Naive Bayes	0.756
flair-news-backward_Naive Bayes	0.737
flair-multi-forward_KNN-3	0.734
flair-multi-forward_KNN-5	0.724
flair-multi-forward_KNN-10	0.686
flair-multi-backward_KNN-3	0.686
flair-multi-backward_KNN-5	0.672
flair-multi-backward_KNN-10	0.629

Top 10 Baselines (Title)

Model	Accuracy
elmo_MLP	0.957
bert_MLP	0.951
elmo_RBF SVM	0.945
elmo_L1 Log Reg	0.942
bert_QDA	0.941
elmo_L2 Log Reg	0.94
bert_RBF SVM	0.939
elmo_Linear SVM	0.939
fasttext_MLP	0.932
bert_Linear SVM	0.929

Bottom 10 Baselines (Title)

Model	Accuracy
flair-news-forward_Naive Bayes	0.769
flair-multi-backward_Naive Bayes	0.756
flair-news-backward_Naive Bayes	0.737
flair-multi-forward_KNN-3	0.734
flair-multi-forward_KNN-5	0.724
flair-multi-forward_KNN-10	0.686
flair-multi-backward_KNN-3	0.686
flair-multi-backward_KNN-5	0.672
flair-multi-backward_KNN-10	0.629
elmo_QDA	0.499

Top 10 Baselines (Concatenation)

Model	Accuracy
bert_MLP	0.951
elmo_L1 Log Reg	0.943
elmo_L2 Log Reg	0.94
bert_RBF SVM	0.939
fasttext_MLP	0.932
bert_L1 Log Reg	0.928
bert_Linear SVM	0.928
flair-news-forward_MLP	0.928
bert_L2 Log Reg	0.927
flair-news-backward_MLP	0.921

Bottom 10 Baselines (Concatenation)

Model	Accuracy
flair-multi-forward_KNN-10	0.686
flair-multi-backward_KNN-5	0.672
twitter_QDA	0.646
flair-multi-backward_KNN-10	0.629
flair-news-forward_QDA	0.624
bert_QDA	0.622
flair-news-backward_QDA	0.611
glove_QDA	0.593
flair-multi-forward_QDA	0.563
flair-multi-backward_QDA	0.547

Top 10 Baselines (Average)

Model	Accuracy
bert_MLP	0.951
elmo_L1 Log Reg	0.943
bert_QDA	0.941
elmo_L2 Log Reg	0.941
bert_RBF SVM	0.939
fasttext_MLP	0.933
bert_Linear SVM	0.929
bert_L1 Log Reg	0.928
bert_L2 Log Reg	0.928
flair-news-forward_MLP	0.926

Bottom 10 Baselines (Average)

Model	Accuracy
flair-multi-forward_Naive Bayes	0.77
flair-news-forward_Naive Bayes	0.769
flair-multi-backward_Naive Bayes	0.756
flair-news-backward_Naive Bayes	0.737
flair-multi-forward_KNN-3	0.734
flair-multi-forward_KNN-5	0.724
flair-multi-forward_KNN-10	0.686
flair-multi-backward_KNN-3	0.686
flair-multi-backward_KNN-5	0.672
flair-multi-backward_KNN-10	0.629