Real Versus Fake News:

NLP Classification With BERT

Amy DeMorrow

CS263

6/7/2022

UCLA MSOL Data Science Engineering

ABSTRACT

Transformers for natural language processing (NLP) have become a standard for high model accuracy metrics on various NLP tasks. Classification is a natural fit for transformer models, and the Hugging Face library has evolved as a popular resource for readily available large scale pre-trained models with great performance on downstream tasks. This project used the Hugging Face "bert-base-uncased" model to classify news articles into "real" and "fake" news. The results show that fine-tuned pretrained models can achieve extremely high accuracy; however, attempting inference with these fine-tuned models on data pulled from different sources, topics, and news cycles yielded poor performance.

INTRODUCTION

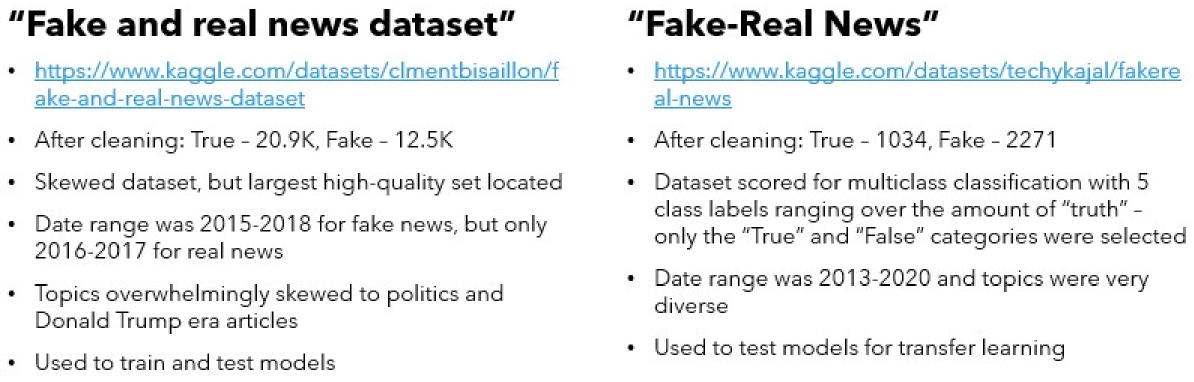
This author has been reading ~2 hours of internet news articles since the late 1990s, so I have a firsthand appreciation of the slow decline of traditional journalism into today's proliferation of "fake" news born from the maturation of the internet landscape. I considered my own personal process for avoiding "fake news" and limiting my consumption of media to "high-quality" news articles, and I wondered if a transformer was powerful enough to learn my tricks.

Perhaps seven years ago, I was still able to infer quality from news title alone, but the vast influx of (what we know understand as) bot activity beginning in 2015-2016 has slowly but surely forced journalists to mimic the bot "click bait" style to compete. Today, almost every article I consider for personal reading uses some form of a short polarizing "click bait" title.

Source has become my number one metric – and those sources continue to shrink. Training on a full text article is also ideal, because I can spot "fake" articles based on the inflammatory and elementary level language alone. As one might expect, high school level sentence structure and content written by journalists has a distinct style that the model can recognize.

The Datasets

My survey of the internet for high quality "real versus fake" news datasets yielded sparse options. Many datasets exist, but few are ideal for transformer-based model training. I narrowed my options to two Kaggle datasets (see Figure 1):



"Fake and real news dataset"

- https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset
- After cleaning: True – 20.9K, Fake – 12.5K
- Skewed dataset, but largest high-quality set located
- Date range was 2015-2018 for fake news, but only 2016-2017 for real news
- Topics overwhelmingly skewed to politics and Donald Trump era articles
- Used to train and test models

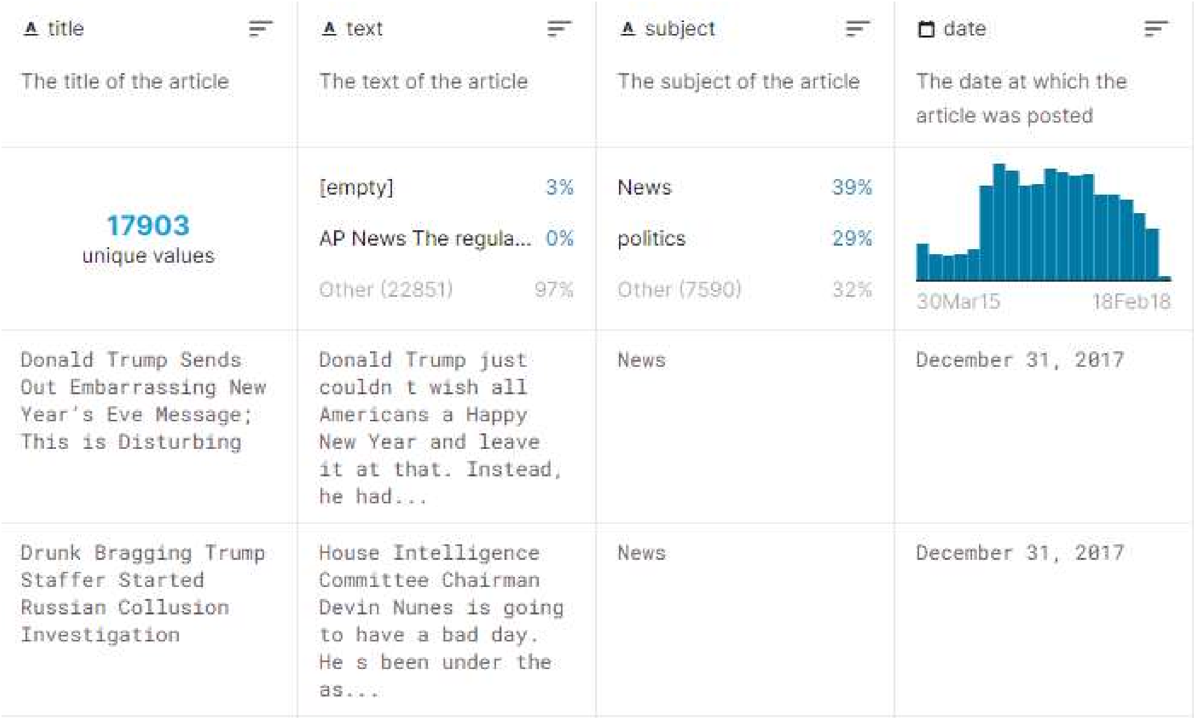
"Fake-Real News"

- https://www.kaggle.com/datasets/techykajal/fakereal-news
- After cleaning: True – 1034, Fake – 2271
- Dataset scored for multiclass classification with 5 class labels ranging over the amount of "truth" – only the "True" and "False" categories were selected
- Date range was 2013-2020 and topics were very diverse
- Used to test models for transfer learning

Figure 1. The datasets used for the BERT binary "real versus fake news" classification task.

The Baseline “Fake and real news dataset”

My baseline dataset was clearly highly skewed (53% of the real news articles and 29% of the fake news articles were political) towards politics and the “Donald Trump” era of the news cycle. The sources were also unclear, although some articles alluded to something Donald Trump posted to Twitter, so these items were not traditional news articles. The lack of clear sources meant this model could not learn from that variable. The dataset was split into two files (“True” and “Fake”) and both files contained four fields – article title, article text, subject, and date (as shown in Figure 2):

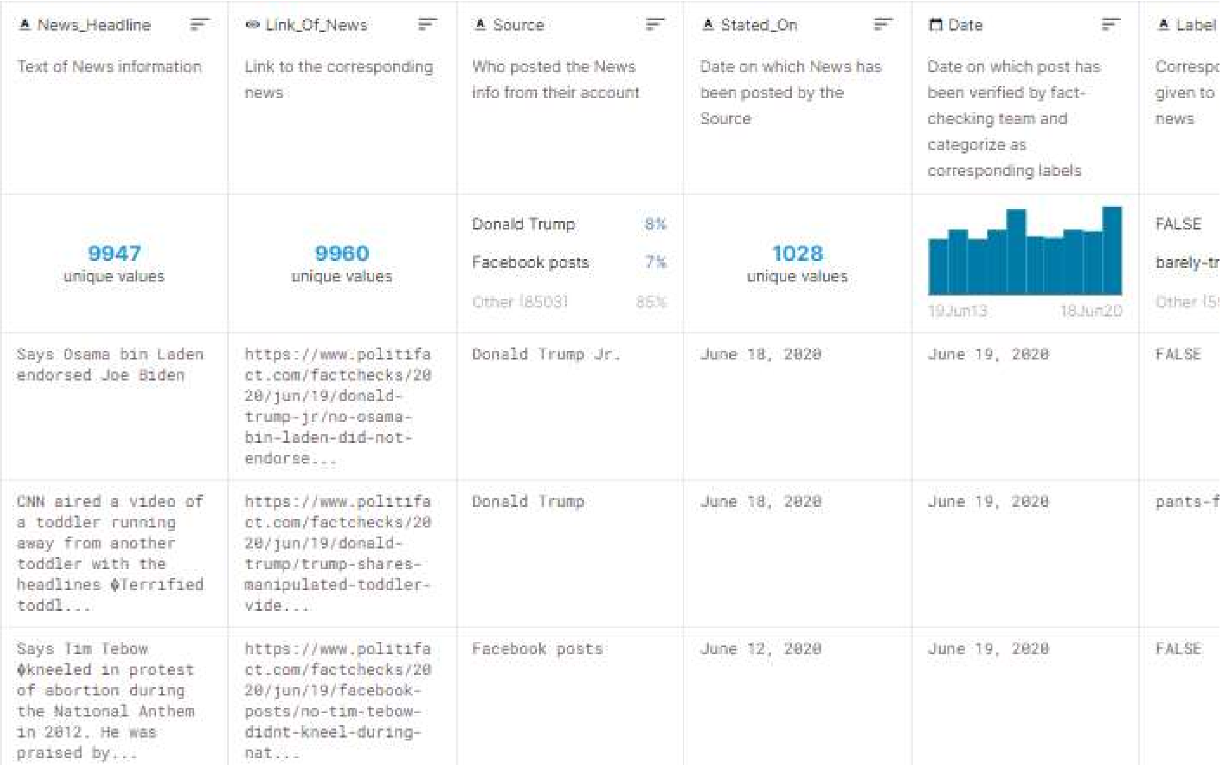


title	text	subject	date
The title of the article	The text of the article	The subject of the article	The date at which the article was posted
17903 unique values	[empty] 3% AP News The regula... 0% Other (22851) 97%	News 39% politics 29% Other (7590) 32%	30Mar15 18Feb18
Donald Trump Sends Out Embarrassing New Year's Eve Message; This is Disturbing	Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had...	News	December 31, 2017
Drunk Bragging Trump Staffer Started Russian Collusion Investigation	House Intelligence Committee Chairman Devin Nunes is going to have a bad day. He s been under the as...	News	December 31, 2017

Figure 2. Format of the “Fake and real news dataset”.

The Inference “Fake-Real News” Dataset

My smaller dataset contained a more diverse category of topics, date ranges, and news sources scraped from the Politifact.com website. The sources were provided, but these “news headlines” were not actual article titles, but simply short statements made by various entities in social media posts, political speeches, and television ads. The lack of full text was a weakness. This dataset was also too small for effective transformer based training, so it was not chosen for baseline model training. This dataset was created for multi-class classification, so I used a subset of only “TRUE” and “FALSE” labels to avoid the “gradient of truth”. This dataset was also highly skewed towards “fake” news, once the “partially true” headlines were eliminated. The dataset is comprised of six fields – news headline, link, source, two dates, and the label (as shown in Figure 3):



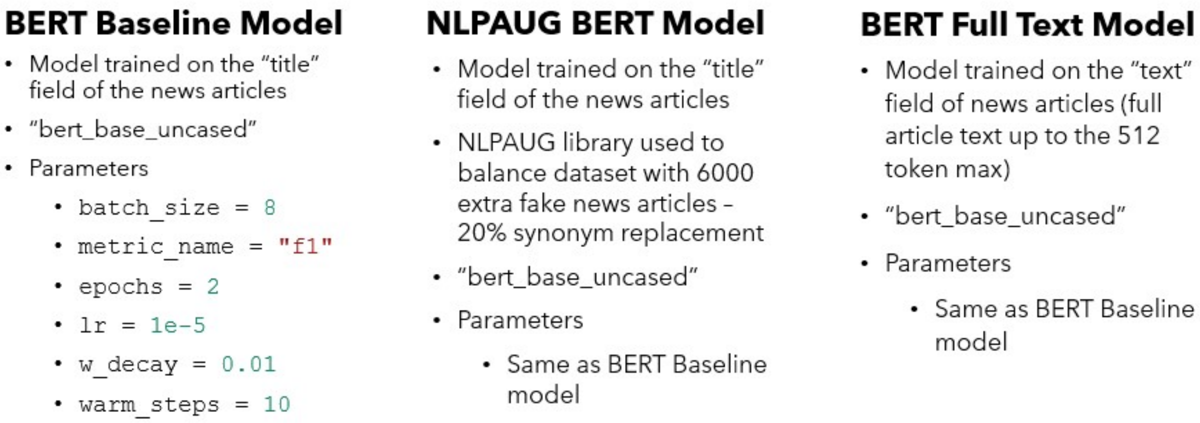
News_Headline	Link_Of_News	Source	Stated_On	Date	Label
Text of News information	Link to the corresponding news	Who posted the News info from their account	Date on which News has been posted by the Source	Date on which post has been verified by fact-checking team and categorize as corresponding labels	Correspo given to news
9947 unique values	9960 unique values	Donald Trump 8% Facebook posts 7% Other (8503) 85%	1028 unique values	19Jun13 18Jun20	FALSE barely-tr Other (5
Says Osama bin Laden endorsed Joe Biden	https://www.politifact.com/factchecks/2020/jun/19/donald-trump-jr/no-osama-bin-laden-did-not-endorse...	Donald Trump Jr.	June 18, 2020	June 19, 2020	FALSE
CNN aired a video of a toddler running away from another toddler with the headlines ϕTerrified toddl...	https://www.politifact.com/factchecks/2020/jun/19/donald-trump/trump-shares-manipulated-toddler-vide...	Donald Trump	June 18, 2020	June 19, 2020	pants-f
Says Tim Tebow ϕkneeled in protest of abortion during the National Anthem in 2012. He was praised by...	https://www.politifact.com/factchecks/2020/jun/19/facebook-posts/no-tim-tebow-didnt-kneel-during-nat...	Facebook posts	June 12, 2020	June 19, 2020	FALSE

Figure 3. Format of the "Fake-Real News" dataset.

The Models

Because I had previously used DistilBERT models, I wanted to try larger and more accurate BERT models for my datasets. I located an on-line article citing a small subset of batch size and learning rate parameters recommended by the BERT developers [1] and common successful weight decays cited were 0.1 and 0.01. I simply carried over the number of warmup steps from the earlier project. My intention was to spend time on data augmentation and model fields rather than fine tuning parameters.

I optimized these parameters on the "title" field of the "Fake and real news dataset" and achieved the highest F1 scores with the parameters listed under the BERT Baseline Model shown in Figure 4. The same parameters were used for an NLPAUG BERT Model and a BERT Full Text Model (details specified in Figure 4) to create a baseline configuration to enable comparing differences in accuracy gained by data augmentation and using more text. Training past epoch 2 showed clear evidence of overfitting, so all model training was halted at epoch 2.



BERT Baseline Model

- Model trained on the "title" field of the news articles
- "bert_base_uncased"
- Parameters
  - `batch_size = 8`
  - `metric_name = "f1"`
  - `epochs = 2`
  - `lr = 1e-5`
  - `w_decay = 0.01`
  - `warm_steps = 10`

NLPAUG BERT Model

- Model trained on the "title" field of the news articles
- NLPAUG library used to balance dataset with 6000 extra fake news articles - 20% synonym replacement
- "bert_base_uncased"
- Parameters
  - Same as BERT Baseline model

BERT Full Text Model

- Model trained on the "text" field of news articles (full article text up to the 512 token max)
- "bert_base_uncased"
- Parameters
  - Same as BERT Baseline model

Figure 4. The three fine-tuned models used for binary classification.

NLPAUG for Data Augmentation

My baseline dataset after cleaning and splitting into the train/dev/test sets was skewed in favor of "true" (label: 1) articles:

```
Train dataset labels count = Counter({1: 14659, 0: 8763})
```

To balance the dataset, I added 6000 additional augmented "fake" (label: 0) article titles using 20% synonym replacement using the NLPAUG library [2]:

```
aug20 = nlpaw.ContextualWordEmbsAug(model_path='bert-base-
uncased', aug_min=1, aug_p=0.2, device="cuda", action="substitute")
```

After augmentation the dataset was well balanced:

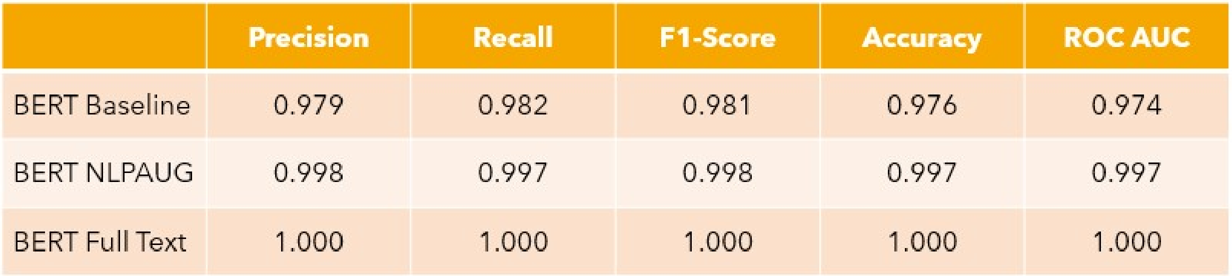
```
Train dataset labels count = Counter({0: 14763, 1: 14659})
```

I preemptively augmented both title and text fields, but accuracy was so high on the full text model, I never trained a model on augmented full text data. Because this augmentation step required almost as much processing time as BERT training, I searched on-line for recommended parameters and a range between 10-20% for the synonym augmentation was suggested. I decided to try the high end and selected 20%. If augmentation time was shorter, I might have attempted 15% and 10%.

RESULTS

Precision, recall, F1, accuracy, and ROC AUC scores were calculated for all three models as shown in Table I below:

Table I. Results of the three models on the "Fake versus real news" test data.

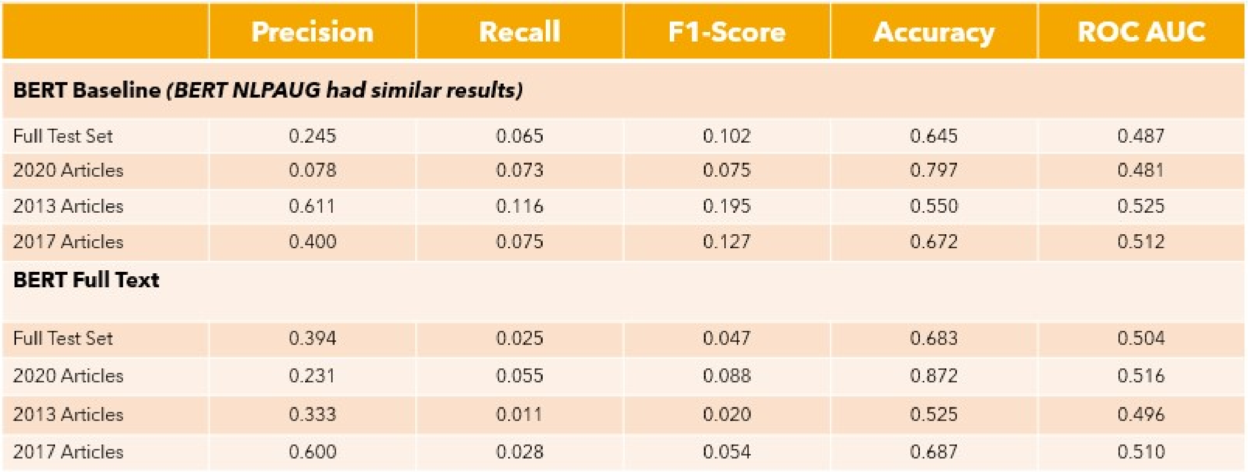


	Precision	Recall	F1-Score	Accuracy	ROC AUC
BERT Baseline	0.979	0.982	0.981	0.976	0.974
BERT NLPAUG	0.998	0.997	0.998	0.997	0.997
BERT Full Text	1.000	1.000	1.000	1.000	1.000

Using BERT for transfer learning yielded very impressive results. All scores on the BERT baseline (title field) exceeded 97%. Using NLPAUG for synonym replacement pushed scores above 99%, which validated that data augmentation on "high quality text" (aka, not Twitter) is a worthwhile use of resources. But results on the full text were most shocking. Multiple training attempts at 45 minutes per attempt never yielded less than 5 out of 5019 misclassified articles (worst score was 0.9997). On the final attempt, the model generated zero misclassified articles, and although that was a fluke, I've reported it as my result just because I enjoyed it so much.

These three fine-tuned models were then used for inference on the second "Fake-Real News" dataset. Although superficially similar (article title versus news headline), the results were extremely poor (see Table II):

Table II. Results of the three models on the "Fake-Real News" dataset.



	Precision	Recall	F1-Score	Accuracy	ROC AUC
BERT Baseline *(BERT NLPAUG had similar results)*					
Full Test Set	0.245	0.065	0.102	0.645	0.487
2020 Articles	0.078	0.073	0.075	0.797	0.481
2013 Articles	0.611	0.116	0.195	0.550	0.525
2017 Articles	0.400	0.075	0.127	0.672	0.512
BERT Full Text					
Full Test Set	0.394	0.025	0.047	0.683	0.504
2020 Articles	0.231	0.055	0.088	0.872	0.516
2013 Articles	0.333	0.011	0.020	0.525	0.496
2017 Articles	0.600	0.028	0.054	0.687	0.510

The results were so shockingly poor, I decided to split the articles into years to see if older and newer headlines (e.g., items outside the temporal range of my original dataset) could be skewing accuracy. Splitting the data out by year didn't reveal a temporal shift – the models are just no better than random chance (and worse) at making predictions. Recall is especially poor with the models predicting the vast majority of items as "fake news".

DISCUSSION/CONCLUSION

For the baseline dataset, using the full text of an article yields best performance. This result makes intuitive sense, because feeding the model more data increases prediction accuracy. The full text model has two additional advantages to aid in prediction – the simplistic language used in fake articles and the location and source tags (`WASHINGTON (Reuters)`) typically present in mainstream news media releases. Although special characters and capital letters were stripped during data cleaning, these location tags were so common, the model couldn't help but learn the pattern. I considered stripping that tag, but realized I use that tag myself to vet the legitimacy of articles. When I see "articles" posted without it I automatically suspect of the validity of the information.

So why did these models perform so well on unseen data in the same dataset, but fail miserably on a different dataset? Although these datasets seemed similar, once I deep dived into the Politifact.com website, I realized that most items posted to that site are "fake", and essentially all the headlines are written as "click bait". Without the experts at PolitiFact and full text available, I would have manually scored most items as "fake", which is essentially exactly how my models behaved. This dataset was intended for training on multiple fields - using headline alone was clearly not enough information for good inference.

I also skimmed both datasets to get a feel for topics. The baseline dataset was highly skewed towards politics (especially Donald Trump), and the smaller dataset was much more diverse spanning from the Obama era of 2013 all the way to the coronavirus era of 2020. Without good topic alignment, these models had little chance of making good predictions, even if the source has been better aligned. Since the topics common to a news cycle shift over time (especially for the bots), models need to be fed new data continuously to maintain high accuracy.

Because news source is so critical to identify "fake news", any future work should involve a multilevel classifier to classify articles first by topic, then by date, and then by full article text and source. In this manner, accuracy would improve simply by sorting coronavirus from politics and Obama from Trump. Using full text and source are ideal to make the final classification into "real" or "fake" - or perhaps even a multiclass classifier since truth seems to exist on a gradient since October 7th, 1996 (I'll let you guys Google that). Regardless of model structure, sorting "real" and "fake" news is a difficult task and requires continuously updated models. The bots and bad actors among us never sleep.

References

[1] Morris, Jack (2020), **“**Does Model Size Matter? A Comparison of BERT and DistilBERT” https://wandb.ai/jack-morris/david-vs-goliath/reports/Does-Model-Size-Matter-A-Comparison-of-BERT-and-DistilBERT--VmlldzoxMDUxNzU

[2] William, Ray (2021), “Hugging Face Transformers: Fine-tuning DistilBERT for Binary Classification Tasks**”,** https://towardsdatascience.com/hugging-face-transformers-fine-tuning-distilbert-for-binary-classification-tasks-490f1d192379

Link to Colab Notebook on Google Drive: https://colab.research.google.com/drive/19e-ZQdDcMbA5PQbTukwXTeYGGRDlZ-KB?usp=sharing