Progress Report - Oct.6

Data Balancing

1. Undersampling - same amount of articles per country

Data Preprocessing

We might adjust this procedure after seeing our actual data:

1. **Remove url and email address**
2. **Remove named entity (NLTK & SpaCy package)**
3. Remove stopwords (NLTK (179 words) & SpaCy (326 words) & Genism (337 words))
4. **Remove special characters**
5. Remove Plural (PorterStemmer) (Not sure if we need this step in BERT)
6. Unify verb tense (WordNetLemmatizer: unify verb tense) (Not sure if we need this step in BERT)
7. Cast to lower

Exploratory Data Analysis

1. Check article distribution across peace & non-peace countries
2. Check average article length before & after preprocessing

Next Step

1. Check to see if current BERT is using lemmetizers or not
2. Practice next steps of preprocessing (removing stop words, etc.) on sample JSON files
3. Do research on how people handle differences between the U.S. and U.K. English language; find out which version of BERT to use
4. Decide whether to use stemer or not
5. Once have data: Make a copy of Philippe's S3 bucket in AWS files into students' own S3 bucket so the original data will not be affected.

Side Notes

Rule-of-5 Countries

High Peace: Australia, New Zealand, Belgium, Sweden, Denmark, Norway, Finland, Czech Republic, Netherlands, Austria

Low Peace: India, Nigeria, Iran, Sri Lanka, Kenya, Congo, Zimbabwe, Uganda, Afghanistan, Guinea

Matched-pairs Countries:

High Peace: Ghana, Chile, Canada, Denmark, Japan, Singapore, Czech Republic, United Arab Emirates

Low Peace: Congo, Brazil, United States, Latvia, China, Philippines, Russia, Iraq

Progress Report - Oct.13

Data preprocessing in BERT preprocessor

- Stopwords are usually kept for BERT, because some stopwords might carry some context, and BERT usually don't need input to be preprocessed with lemmatizing & stemming due to the same reason.
  Ex: not, nor, never are stopwords, but have strong contextual meanings.

- BERT cased
  - Tokens for a sentence with capital letters is not the same as tokens for the same sentence all in lowercase
    Ex: us != US ; for us meaning we, and US meaning the country or the currency name like US$
- BERT uncased
  - Convert everything to lowercase first, then do the tokenization

  -> We don't need 'Convert to lowercase' in our own preprocessing step since the BERT will handle it depending on which version we are going to use.

- BERT has a WordPiece Model which can handle the subword tokenization as following
  Ex: embeddings -> 'em', '##bed', '##ding', '##s'
  Note: Some words might have different word embeddings for different tense:
  'announce' vs. 'announced' are both in the BERT word bank

Practice next steps of preprocessing (removing stop words, etc.) on sample JSON files

- We will update the code and upload the file to GitHub when we have access to the data

Do research on how people handle differences between the U.S. and U.K. English language; find out which version of BERT to use

- Option 1: Convert U.K. English to U.S English and then do the U.S English BERT (or ALBERT, ELECTRA, etc)

Reference:
https://blog.tensorflow.org/2020/12/making-bert-easier-with-preprocessing-models-from-tensorflow-hub.html

Next Step

1. Can you run BERT on AWS? Is there a way to estimate the computation?
2. Take some samples from full dataset, in a logarithmic way, i.e. 1,2,4,8 (not arithmetic or linear way i.e. 1,2,3,4, megabytes) increasing it by a factor of 10 or 2 every time, running on a few small samples of data so we can get an estimate of run times and expense
3. Find which pretrained model is appropriate to use on the data we have

Progress Report - Oct 20

Run BERT in SageMaker

- Using HuggingFace with AWS SageMaker is more commonly seen than using Tensorflow, but both packages are available on SageMaker.
- AWS has a direct collaboration with HuggingFace -> Easy-to-use package to load & fine-tune in any pre-trained BERT models on HuggingFace.
- HuggingFace Models usually can be loaded as Tensorflow layers (haven't checked if this is true on SageMaker yet).
    - If it is true, then we can build only one model pipeline: use the Tensorflow package on SageMaker for the model, then insert the HuggingFace BERT model as tf layer.
    - Otherwise, we can try each package separately.

Helpful Tutorials

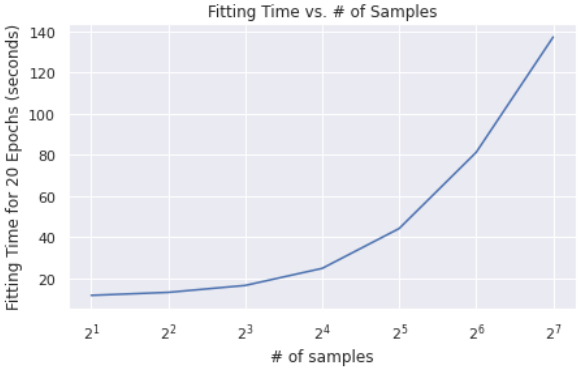
- Use BERT from HuggingFace:
    - Train a Hugging Face Transformers Model with Amazon SageMaker
- Use BERT from Tensorflow: helpful and short tutorial for beginners, covers loading data from s3 bucket, preprocessing it, storing the result file back to s3, then start a tensorflow model; might be easier to understand than the first tutorial)
    - Part 1 - Deploy TensorFlow Models on Amazon AWS SageMaker
    - Part 2 (final) - Deploy TensorFlow Models on AWS SageMaker

AWS Sagemaker Official Tutorial

https://aws.amazon.com/getting-started/hands-on/build-train-deploy-machine-learning-model-sagemaker/

BERT Computation Time Estimate

- BERT-base-uncased + one output dense layer on Colab GPU
- Average the runtime of 5 trials, 20 epochs per trial



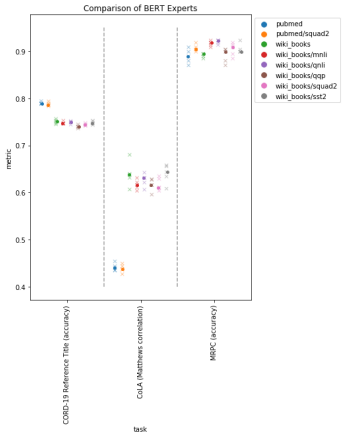
Find which pretrained model is appropriate to use on the data we have

Model	Platform	Dataset	Model Size	Vocab Size	Other Info
RoBERTa-base	HuggingFace	Pretrained on: • BookCorpus, a dataset consisting of 11,038 unpublished books • English Wikipedia (excluding lists, tables and headers) • CC-News, a dataset containing 63 millions English news articles	12-layer, 768-hidden, 12-heads, 125M parameters (vs. 110M parameters in BERT-base)	50k	SST-2 Score: 94.8 MRPC: 90.2

		crawled between September 2016 and February 2019 • OpenWebText, an open source recreation of the WebText dataset used to train GPT-2 • Stories a dataset containing a subset of CommonCrawl data filtered to match the story-like style of Winograd schemas.			
experts/bert/wiki_books/onli	Tensorflow	(Fine-tuned version of BERT-base-uncased) Pretrained on: • English Wikipedia and BookCorpus Fine-tune: • QNLI Task	110M	30k	Best performance on MRPC among other models fine-tuned for different tasks: Score of 90+
textattack/distilbert-base-uncased-ag-news	HuggingFace	Pretrained on: • Distill-BERT has a pretrain dataset same as BERT(English Wikipedia and BookCorpus) Fine-tune: • Adversarial Attack Approach on News Category Classification	66M	30k	ag_news sequence classification: 94.7

Reference:

All Tensorflow BERT Experts Performance



CORD-19 task: COVID-19 Open Research. Search question answers among a large set of COVID related scientific papers

CoLA task: The Corpus of Linguistic Acceptability. Classify sentences as grammatical or not grammatical

MRPC task: sentence semantic comparison & classification on online news articles.

- The Microsoft Research Paraphrase Corpus (MRPC) is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent.

QNLI task: determine whether the context sentence contains the answer to the question.

SST-2 task : binary sentiment analysis/classification on IMDb movie reviews

Question:

1. Should we preprocess all of the data and save the results in a s3 bucket or should we put the preprocessing steps in the model?
    - **Pros** for saving results:
        - Save time for model training, reduce the cost on GPU usage
        - Only have to do this once, and the result would be useful for all BERT models we'd like to explore.
    - **Cons** for saving results:
        - Need more s3 bucket resources.
    - **Solution:** Saving data to S3 Storage is much cheaper. But we need to know how much data will be generated from the pipeline in order to set up the best AWS service.