Capstone Datasets Explored

1)

[TensorFlow 2.0 Question Answering | Kaggle](https://www.kaggle.com/competitions/tensorflow2-question-answering/data)

Dataset files: v1.0-simplified\_simplified-nq-train.jsonl.gz

columns/fields

dict\_keys(['annotations', 'document\_html', 'document\_title', 'document\_tokens', 'document\_url', 'example\_id', 'long\_answer\_candidates', 'question\_text', 'question\_tokens'])

Nores: It has answer candidates but not labels.  Is this meant to use other pretrained models?

It is the HTML from wiki pages.  They give start and end character locations for the candidates.

2)

[Google QUEST Q&A Labeling | Kaggle](https://www.kaggle.com/competitions/google-quest-challenge)

Dataset files: google-quest-challenge.zip

In this competition, you’re challenged to use this new dataset to build predictive algorithms for different subjective aspects of question-answering. The question-answer pairs were gathered from nearly 70 different websites, in a "common-sense" fashion. Our raters received minimal guidance and training, and relied largely on their subjective interpretation of the prompts. As such, each prompt was crafted in the most intuitive fashion so that raters could simply use their common-sense to complete the task. By lessening our dependency on complicated and opaque rating guidelines, we hope to increase the re-use value of this data set. What you see is what you get!

Columns/fields

Index(['qa\_id', 'question\_title', 'question\_body', 'question\_user\_name',

       'question\_user\_page', 'answer', 'answer\_user\_name', 'answer\_user\_page',

       'url', 'category', 'host', 'question\_asker\_intent\_understanding',

       'question\_body\_critical', 'question\_conversational',

       'question\_expect\_short\_answer', 'question\_fact\_seeking',

       'question\_has\_commonly\_accepted\_answer',

       'question\_interestingness\_others', 'question\_interestingness\_self',

       'question\_multi\_intent', 'question\_not\_really\_a\_question',

       'question\_opinion\_seeking', 'question\_type\_choice',

       'question\_type\_compare', 'question\_type\_consequence',

       'question\_type\_definition', 'question\_type\_entity',

       'question\_type\_instructions', 'question\_type\_procedure',

       'question\_type\_reason\_explanation', 'question\_type\_spelling',

       'question\_well\_written', 'answer\_helpful',

       'answer\_level\_of\_information', 'answer\_plausible', 'answer\_relevance',

       'answer\_satisfaction', 'answer\_type\_instructions',

       'answer\_type\_procedure', 'answer\_type\_reason\_explanation',

       'answer\_well\_written'],

      dtype='object')

In [ ]:

​

3)

[U.S. Patent Phrase to Phrase Matching | Kaggle](https://www.kaggle.com/competitions/us-patent-phrase-to-phrase-matching/data)

Dataset file: us-patent-phrase-to-phrase-matching.zip

In this dataset, you are presented pairs of phrases (an anchor and a target phrase) and asked to rate how similar they are on a scale from 0 (not at all similar) to 1 (identical in meaning). This challenge differs from a standard semantic similarity task in that similarity has been scored here within a patent's context, specifically its [CPC classification (version 2021.05)](https://en.wikipedia.org/wiki/Cooperative_Patent_Classification), which indicates the subject to which the patent relates. For example, while the phrases "bird" and "Cape Cod" may have low semantic similarity in normal language, the likeness of their meaning is much closer if considered in the context of "house".

This is a code competition, in which you will submit code that will be run against an unseen test set. The unseen test set contains approximately 12k pairs of phrases. A small public test set has been provided for testing purposes, but is not used in scoring.

Information on the meaning of CPC codes may be found on the [USPTO website](https://www.uspto.gov/web/patents/classification/cpc/html/cpc.html). The CPC version 2021.05 can be found on the [CPC archive website](https://www.cooperativepatentclassification.org/Archive).

Index(['id', 'anchor', 'target', 'context', 'score'], dtype='object')

4)

[UNIFESP X-ray Body Part Classifier Competition | Kaggle](https://www.kaggle.com/competitions/unifesp-x-ray-body-part-classifier/data)

Dataset file: unifesp-x-ray-body-part-classifier.zip

Classifying a body part from an x-ray image might seem silly, but having it automated can be a key for all the world around deep learning in medical imaging. In many hospitals, when a physician orders multiple imaging exams one accession number is created for each body part (eg. knee, ankle, and leg), but the registration for the correspondent images are often incorrect within each accession number, here is an example:

In this challenge, competitors are predicting the body part from a given x-ray.

When making predictions, competitors should predict as many body parts per image as they judge necessary

There should be 1 predicted column per image - and the labels are represented as integers that map each to one body part contained in the dataset.

A properly formatted row may look like any of the following.

SOPInstanceUID, 0 12

The labels are represented as integers that map to the following:

* Abdomen = 0
* Ankle = 1
* Cervical Spine = 2
* Chest = 3
* Clavicles = 4
* Elbow = 5
* Feet = 6
* Finger = 7
* Forearm = 8
* Hand = 9
* Hip = 10
* Knee = 11
* Lower Leg = 12
* Lumbar Spine = 13
* Others = 14
* Pelvis = 15
* Shoulder = 16
* Sinus = 17
* Skull = 18
* Thigh = 19
* Thoracic Spine = 20
* Wrist = 21

Note - Others indicates whether the sample contains image of non X-ray images that sometimes are misplaced in the PACS system as X-Ray (eg. esophagogram, densitometry).