# 資訊檢索與擷取 Generative Information Retrieval

## HW2: Fact Checking

TA: Hank Chen

h7a4n1k.cs12@nycu.edu.tw

# Task Description

**Topic:** Fact Checking

Nowadays, we obtain information from a lot of different sources, either from news, social media, forums, etc. However, it is unwise to blindly believe everything you see or hear. This is where fact checking comes into play. Given several **reference articles** and a **claim**, the task is to determine whether the claim is **true** or **false**.

# Task Description

**Goal:** Accurately classify each claim to be “True”, “Partial True” or “False”.

1. Submit a report and source code to E3
2. Upload your prediction to Kaggle competition

# Dataset

The dataset contains **3 files** and 1 directory:

- **train.json / valid.json / test.json**:
  - Each json file is a list that contains multiple claim objects, which stores the metadata and label of a claim. (Note: test set has no label)

# Dataset

- **metadata** - definition of articles
  - **claimant** - the person who claims.
  - **claim** - the claim
  - **id** - claim id
  - **premise_articles** - {} of urls, and the names of provided json files
- **label**
  - **rating** - 0: false; 1: partial true; 2: true
  - **original_rating** - original sentence to clearify rating
  - **id** - claim id

# Dataset Example


```
"metadata" : { 4 items
  "claimant" : string "Joe Biden"
  "claim" : string "“38,000 prisoners were released from federal prison” during the Obama administration."
  "id" : int 2
  "premise_articles" : {...} 87 items
}
"label" : { 3 items
  "rating" : int 1
  "original_rating" : string "misleading"
  "id" : int 2
}
```


There are a lot of reference articles!

# Dataset

The dataset contains 3 files and **1 directory**:

- **articles/**
  - This directory contains 258k json files. Each json file is web crawled content of the url in the metadata. Note that **the quality of these articles is quite low**. You will need to preprocess the data.

# Dataset Example


```
"premise_articles" : { 87 items
  "
  https://www.facebook.com/sharer/sharer.php?
  u=https%3A%2F%2Fwww.factcheck.org%2F2020%2F10%2Ffactchecking-the-final-2020-presidential-
  debate%2F%3Futm_source%3Dfacebook%26utm_medium%3Dsocial%26utm_campaign%3Dsocial-
  pug&t=FactChecking%20the%20Final%202020%20Presidential%20Debate
  "
   : string "2_1.json"
  "
  https://twitter.com/intent/tweet?
  text=FactChecking%20the%20Final%202020%20Presidential%20Debate&url=https%3A%2F%2Fwww.factcheck.org%2F
  the-final-2020-presidential-debate%2F%3Futm_source%3Dtwitter%26utm_medium%3Dsocial%26utm_campaign%3Ds
  pug&via=factcheckdotorg
  "
   : string "2_2.json"
  "
  https://www.tumblr.com/widgets/share/tool?
  canonicalUrl=https%3A%2F%2Fwww.factcheck.org%2F2020%2F10%2Ffactchecking-the-final-2020-
  presidential-debate%2F%3Futm_source%3Dtumblr%26utm_medium%3Dsocial%26utm_campaign%3Dsocial-pug
  "
   : string "2_3.json"
```

# Kaggle

- Competition Link
- Create a team with your **student ID**, we use this information for grading. Your team name should be exactly **<student_ID>**.
- If you failed to do so under any circumstances, there will be **a penalty of 10 points** to your score, so be sure to use the correct team name.

# Kaggle (cont.)

- Public leaderboard is calculated with 50% of the test set, private leaderboard is calculated with the other 50%, the final standings may be different.
- Therefore, **please DO NOT overtune your model to fit the public leaderboard, or you will suffer from overfitting**.

# Kaggle (cont.)

- The scoring metric is **macro-F1**.
- We have set a **simple baseline** and a **strong baseline**, beat them to get higher score.
- You can submit at most 10 times each day and choose 2 of the submissions to be scored for the private leaderboard, or will otherwise default to the best public scoring submissions.

# Kaggle Submission Format

- Report claims with their **id (attribute “id” in each claim object)**, and their corresponding prediction.
- There should be **2361 x 2** entries in your csv file, **with columns “id” and “rating” (0: False; 1: Partial True; 2: True)**.
- The order of the ids does not matter. Refer to sample_submission.csv for the correct format.

# Possible Approaches & Tips

- **Disclaimer:** The following appoarches are just for your reference. You are not required to follow any of them. We encourage you to invent new ways to solve this problem.

# Possible Approaches & Tips

- The challenging part of this task are, **(1) preprocess and retrieve useful and structured information from the premise articles**, and **(2) predict the claim with retrieved information.**

# Possible Approaches & Tips

- **Preprocessing and Retrieval**
  - Use information retrieval techniques (e.g. TF-IDF, BM25, BERTScore, ...) to find **relevant articles about the claim**.
  - These articles are provided in the sentence level, use IR to find **evidence sentences** that may be helpful for prediction.

# Possible Approaches & Tips

- **Claim Prediction**
  - Use collected evidence sentences to
    - train a sequence-to-sequence model (e.g. BERT, RoBERTa, BART, ...) that can help you classify the claim
    - or use a Large Language Model to do the prediction.

## Report

Please answer the following questions, provide your thinking process as detailed as possible:

1. (10%) Describe your approach to **data proprocessing and information retrieval**. Please choose **at least 2 of any IR methods** and **compare their performance**.

# Report (cont.)

2. (10%) Describe your approach to **claim prediction**. Details such as model selection, hyperparameters **should be provided**.

If you use LLMs, try using techniques such as in-context learning, chain-of-thought prompting or any techniques that can improve the quality of generation and **compare their performance**.

# Report (cont.)

3. (10%) Do **error analysis** or **case study**. Is there anything worth mentioning while checking the mispredicted data? Share with us. Anytime you try to make a conclusion about the data or model, you should provide concrete data example.

[e.g. I think the model predicts poorly when ... (provide error examples)]

**There is no “correct answer” to above questions, just do your best and answer them in detail.**

# Grading Policy

- **Kaggle (70% of total)**, details in the next slides.
- **Report (30% of total)**, points are attached at the start of each question.

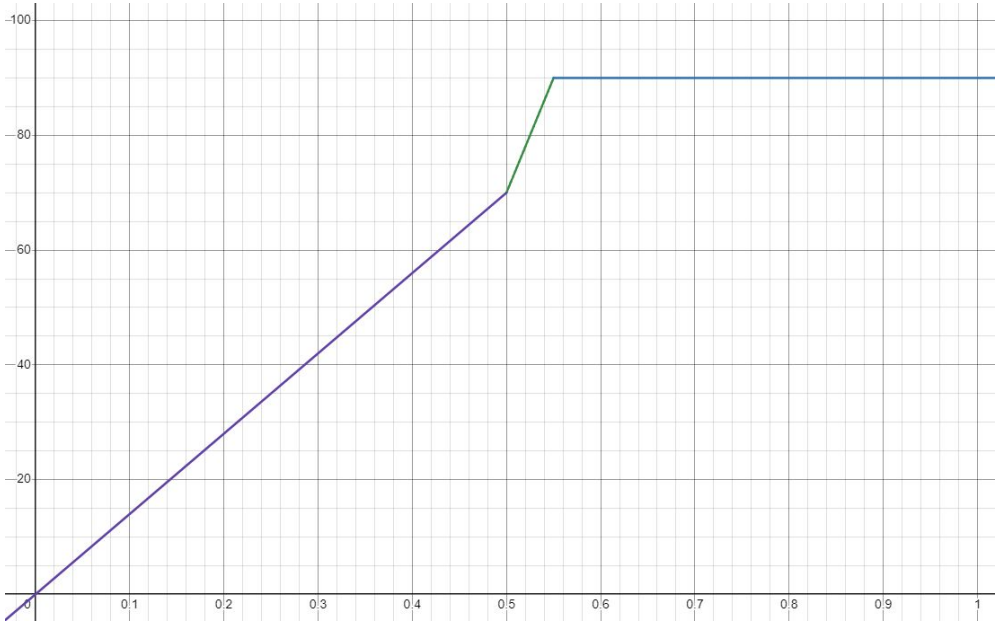
# Grading Policy - Kaggle

- **Baselines (90% of Kaggle; 30% public, 70% private)**
  - We have 2 baselines: Strong Baseline (**B_1**), Simple Baseline (**B_2**)
  - **Your score > B_1:** 90
  - **B_1 > Your score (X) > B_2:** (X - B_2) / (B_1 - B_2) * 20 + 70.
  - **Your score (X) < B_2:** X / B_2 * 70.

# Grading Policy - Kaggle

- Assume that:
  - **B_1** = 0.55
  - **B_2** = 0.5

- x-axis: F1
- y-axis: score



# Grading Policy - Kaggle

- **Ranking (10% of Kaggle)**
  - Compete with your classmates.
  - Rank 1-5: 10%
  - Rank 6-10: 8%

    ...

  - Rank 21-25: 2%
  - Rank > 25: no points :(

# E3 Submission

- Submit your source code and report to E3 before **11/26(Tue.) 23:59**, no late submissions will be accepted.
- Please submit your source code in **python source (.py)**. For jupyter notebooks, you can use the export function to obtain the executable script.
- **Do NOT put full code in your report.** Only short code snippets and pseudocode (for demonstration purposes) are allowed, and they should be properly formatted.

# E3 Submission

**Submission format:**

- hw2_<student_id>.zip
  - **source code:** hw2_<student_id>.py or other library files (.py) you made
  - **report:** hw2_<student_id>.pdf

- Failed to comply with above rules (under any circumstances) will cause a **deduction of 5 points** to your score.

# Contact

- If you have any questions about Homework 2, please feel free to contact with TA by email:
  - Hank Chen (h7a4n1k.cs12@nycu.edu.tw)

- **Kindly format your email** so that I can reply it ASAP.
  - **Prefix your email title with [IRIE]**. (e.g. [IRIE] Questions about Homework 2)
  - Attach your student ID if necessary.