**What to ask the TA**

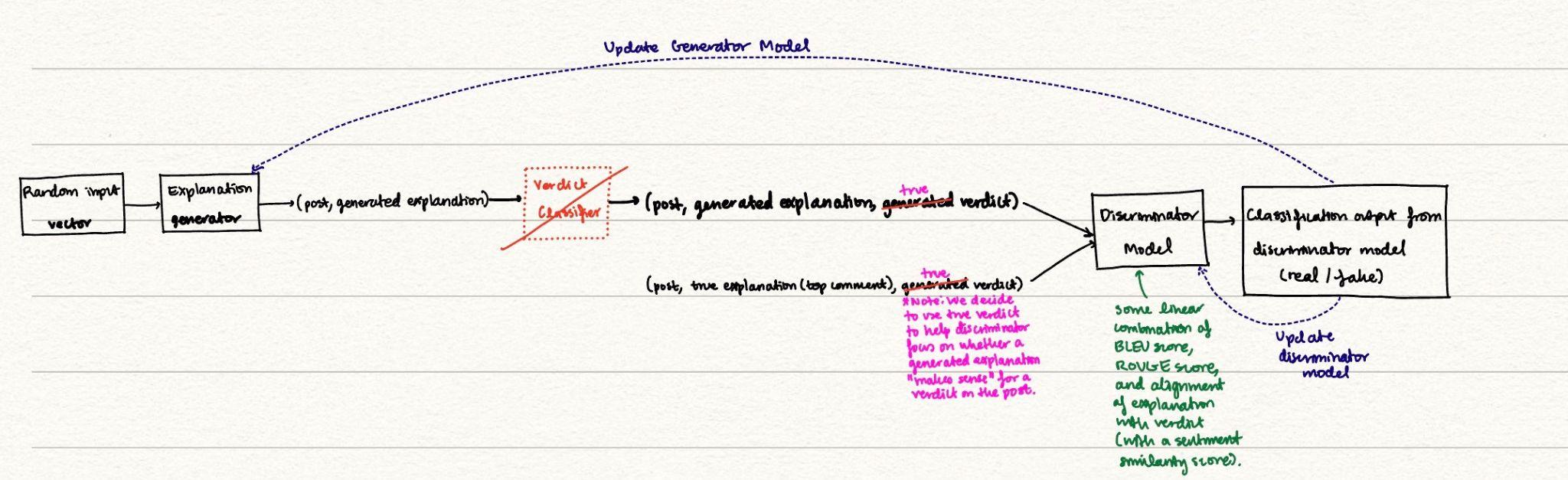
* How to incorporate the verdict itself as external information?
  + Should it be **true** or generated?
  + Turn it into 1 hot encoding
* What do we pretrain our discriminator on so that it’s not too strong but also not incompetent?
* What kind of model is appropriate for the generator? Does it have to be Seq2Seq/Transformer or can we use other models like VAE?
* Should we incorporate BLEU and ROUGE for loss computations in the Discriminator?
* Our ideas for baselines:
  + (One more baseline)
  + DistilBERT
  + VAE + RNN like Transformer (b/c our final model might be [VGAN](https://arxiv.org/pdf/1712.00170.pdf))

**Notes from 2nd TA meeting on Fri Mar 3**

* Approach put the whole picture
* And future works put the discriminator
* Other metrics for language generations: cider spice meter (different ways to quantify similarity). BLEU is n-gram similarity.
* Coco captions – generation. Python metrics for language generation coco.

Include a figure for your method. Give general dataset metrics. Num samples, avg length. Num Examples for each class. Include train/test split. Add subject tags (maybe). Keep distribution within each split. Consider up sampling the minority classes.

**Our model**

****

* Uses adversarial training to generate an explanation for the verdict
* Verdict included in training set, as latent variable in GAN, to generate better explanations
* Explanation generator
* Verdict classifier
  + Trained on (post, explanation)
  + Output verdicts go to discriminator
* Discriminator
  + Distinguishes between generated and true tuples (post, explanation, verdict)

**TA Thoughts**

DistilBert will generate very short answers. Don’t recommend it because we want a longer explanation.

Baseline – train an autoregressive decoder. Seq2Seq architecture. Decoder can be a transformer model. Or if you find RNN easier to implement, you can try that. Post is the input. Explanation as the output for the generator.

* Hugging face T5 small (pretrained so it has external )
  + Pretrained and not pretrained
* Text2Text model. Or conditional generator.
* GPT 2 (autoregressive decoder) as another baseline.
  + Can specify post as context parameter.

This is not really a question answering problem. Because the answer can be found in the question. This is like a new task. Not fully reading comprehension either.

Typical Discriminator can be simple like RNN or simple layer transformer. Can try LSTM or transformers.

* Option 1: Can add an autoregressive loss. You have a baseline and add in a GAN out of the baseline
  + Teacher forcing style to make sure generator continues making fluent sentences
  + Don’t care about BLEU
* Option 2: Use BLEU but figure out how to back prop the BLEU score.
  + Just like negative log likelihood. Will function like option 1

**Key Info**

**Abstract**

**Approach**

* Include key equations and figure, what the baselines are. Describe original parts of approach and parts copied. Describe code you wrote vs copied.

**Experiments**

* Datasets
* Evaluation method
* Experimental details
* Results
  + Comment on quantitative results

**Future work**

**References**