Hi! I am Prathamesh. I’ll be explaining my work briefly in this doc

The Task

Given a conversation transcript between an agent and a client, output the date when the person will make the next payment. If it cannot be inferred the output ‘NA’.

Sample data point:

{
"id": 112,
"conversation_date": "2020-01-20, Monday",
"conversation": "Agent: When should we expect the payment?\nCustomer: I can make the payment by the end of the month.",
"label": "2020-02-28",
"days_diff": 39
}

Approaches I explored

- **Algorithm based:** I didn’t try this approach as it would defeat the purpose of this assignment (i.e. testing proficiency in ML/AI/NLP)
- **Entity recognition:** We could use a very basic model based on entity recognition (scipy - en_core_web_lg) but this is unable to understand semantic complexity of dialogue exchange and hence we will also need deep learning.
- **Hidden attribute models:** Ham uses attention to form <person, attribute, value> but we'd need a bigger dataset to train on for this to work.
- **Using an API like ChatGPT:** Just putting it here because in my experiments I got pretty much 95%+ accuracy using this.
- **Entity recognition + NLP models:** Since I don’t have a very big dataset to finetune/train a model on, I’ll be using a big generalized pretrained model

My Final Approach

- **Dataset Creation:**
  - I generated a ~100-sample dataset with ChatGPT and then manually curated it (chatgpt_gen_date.json). This will be my train set.
  - I also used ChatGPT to generate labels for the given sample set (by Salient) of ~110 samples and also reformatted it. This will be my test set. (data/test_data.json)
- **Finetuning**
  - I trained 2 types of models.
    - Target is the date in the form yyyy/mm/dd
    - Target is the number of days after the call's date that the user will pay the amount.
  - 3. I finetuned various variants of FlanT5 (i.e. small, base, large). Bigger models performed better as expected. I was unable to try the xl and xxl variants because of compute & time restrictions.

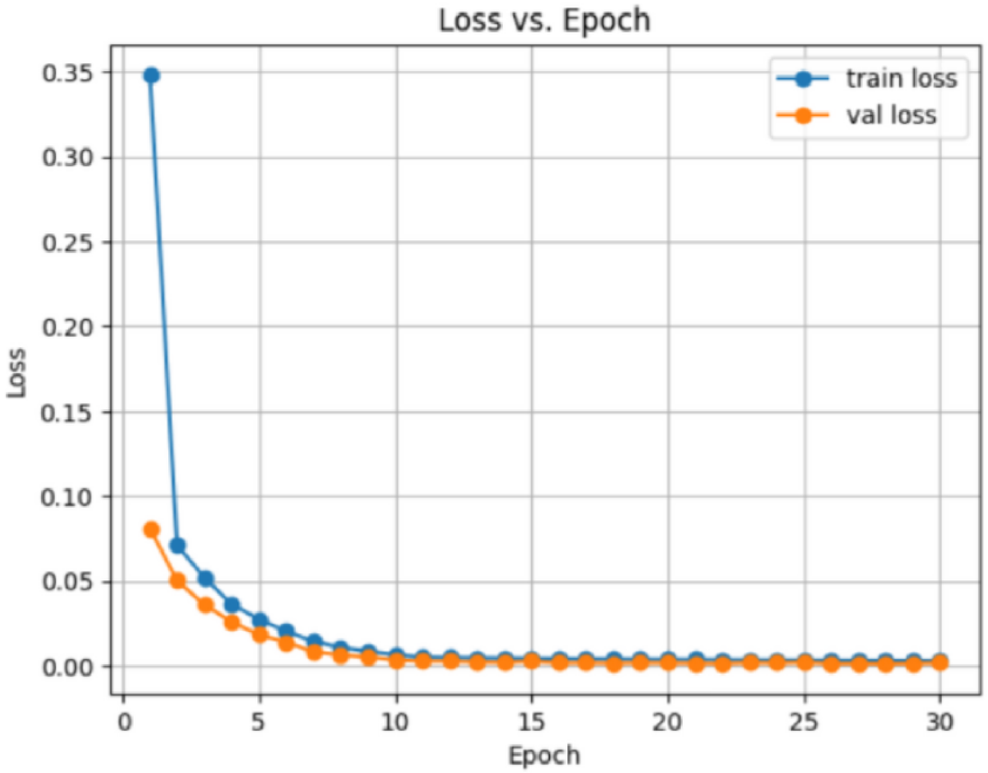
My Final Approach (Continued)

- **Entity recognition:**
  - If the data doesn’t contain 'ORDINAL' or 'DATE' entities then there is no mention of any time/date/etc in the conversation and we can safely return NA. Using this trick is useful as I was instantly able to process 20/113 responses as NA. False positives = 0.
  - Pass the conversation along with the prompt to get the output.
  - OPTIONAL: Ensembling - I have also created an ensemble of both types of models above to get a more reliable output.
- **Evaluation**
  - Exact matching accuracy
  - Standard deviation
  - Rouge1, Rouge2, Rougel, Rougelsum

My best model

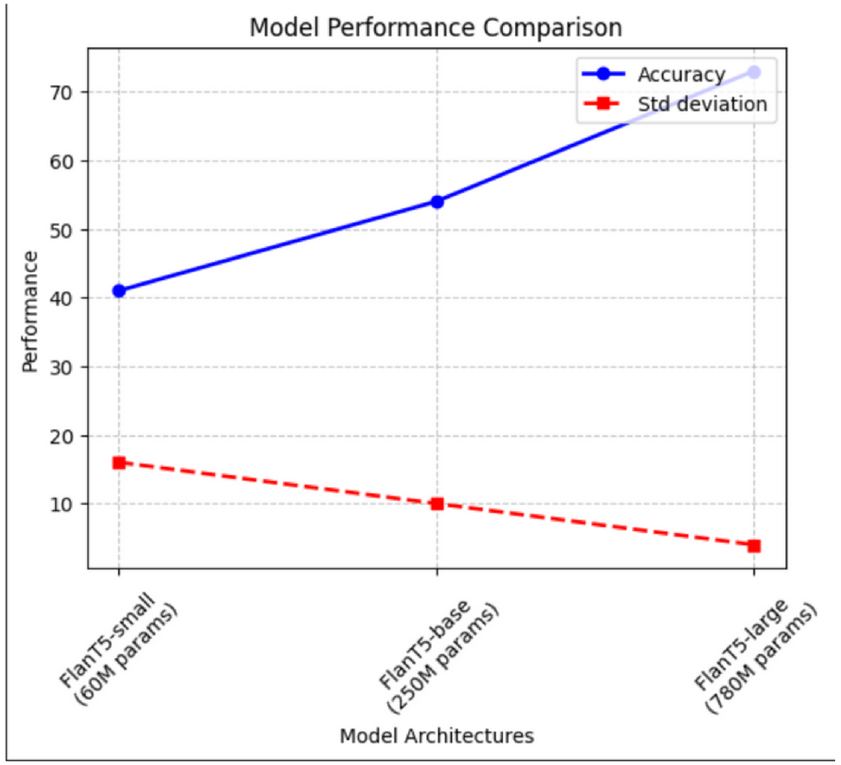
Flan T5 - large (780M params) finetuned on 130 samples

Validation accuracy = 73%
std deviation = 4.12
on 100 held out samples.



Performance

Although due to time and compute limitation I could try larger models, I evaluated this on smaller model architectures.



Conclusion:

- One major drawback for these experiments was dataset size. To get a more solid set of metrics we'd need a bigger dataset.
- The models performed better as the architecture sizes increased. Observing the trend we'd probably get very good accuracy with bigger models like Vicuna, FlanT5-xxl.
- Since the dataset was small we were only able to finetune it a bit but given a bigger dataset we can finetune it in a much better fashion or even train the whole model. Alternatively, we could also train a model from scratch given enough data. (this model could be much smaller)
- In general the latency at which we got the response from these models was under 0.3 seconds. Following the industry standard of 2s response time, this is a good number.