Making Machine Converse Better

Deepak Goyal   Deepak Agrawal   Nitanshi Mahajan

Prof. Subrata Bhattacharya

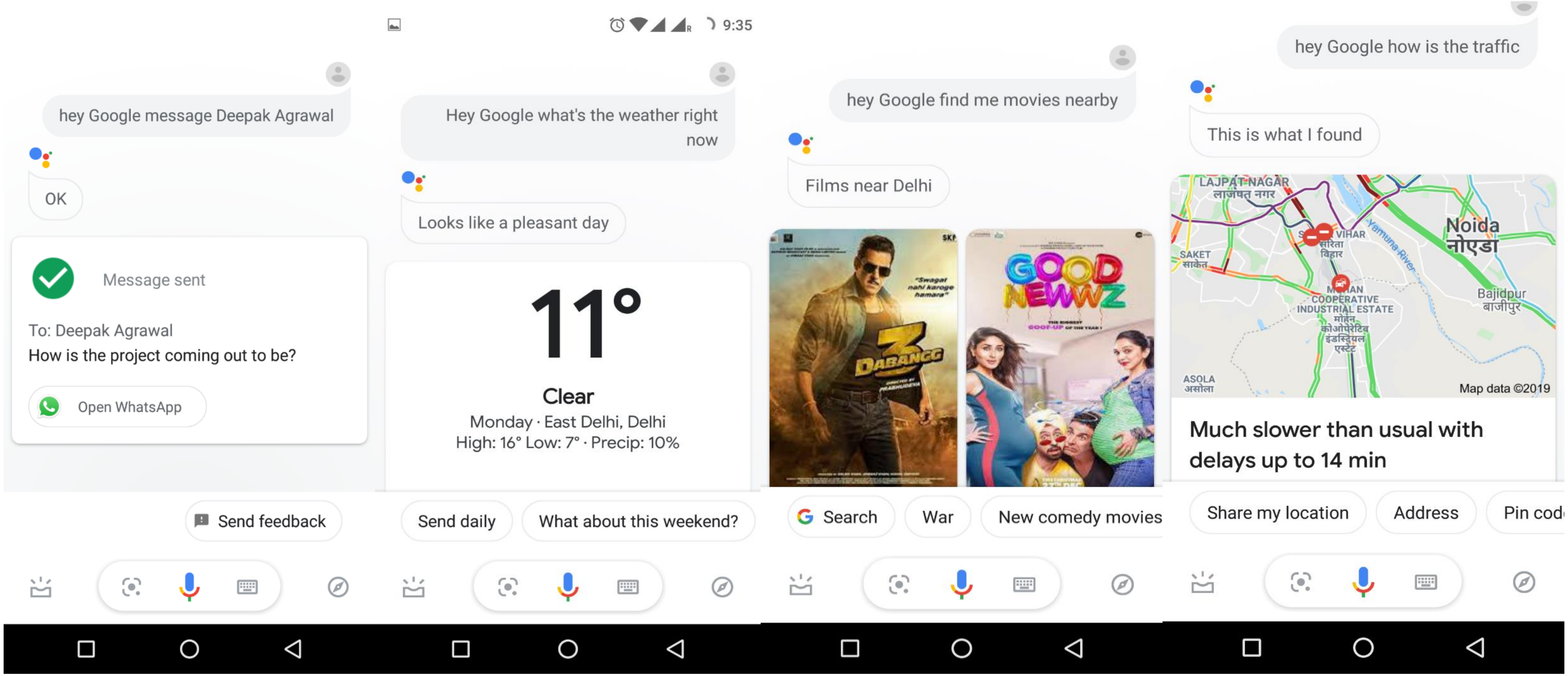
Overview

- The problem of Open-Ended Conversational Agents[5 minutes]
- Acknowledging the problem through experiments[5 minutes]
- The problems and the solutions[5 minutes]
- Solution Part 1: Making Better Models[5 minutes]
- Solution Part 2: Including Context and Modelling Explicit Information[5 minutes]
- Solution Part 3: Adding Control[5 minutes]

Open-Ended Conversational Agents

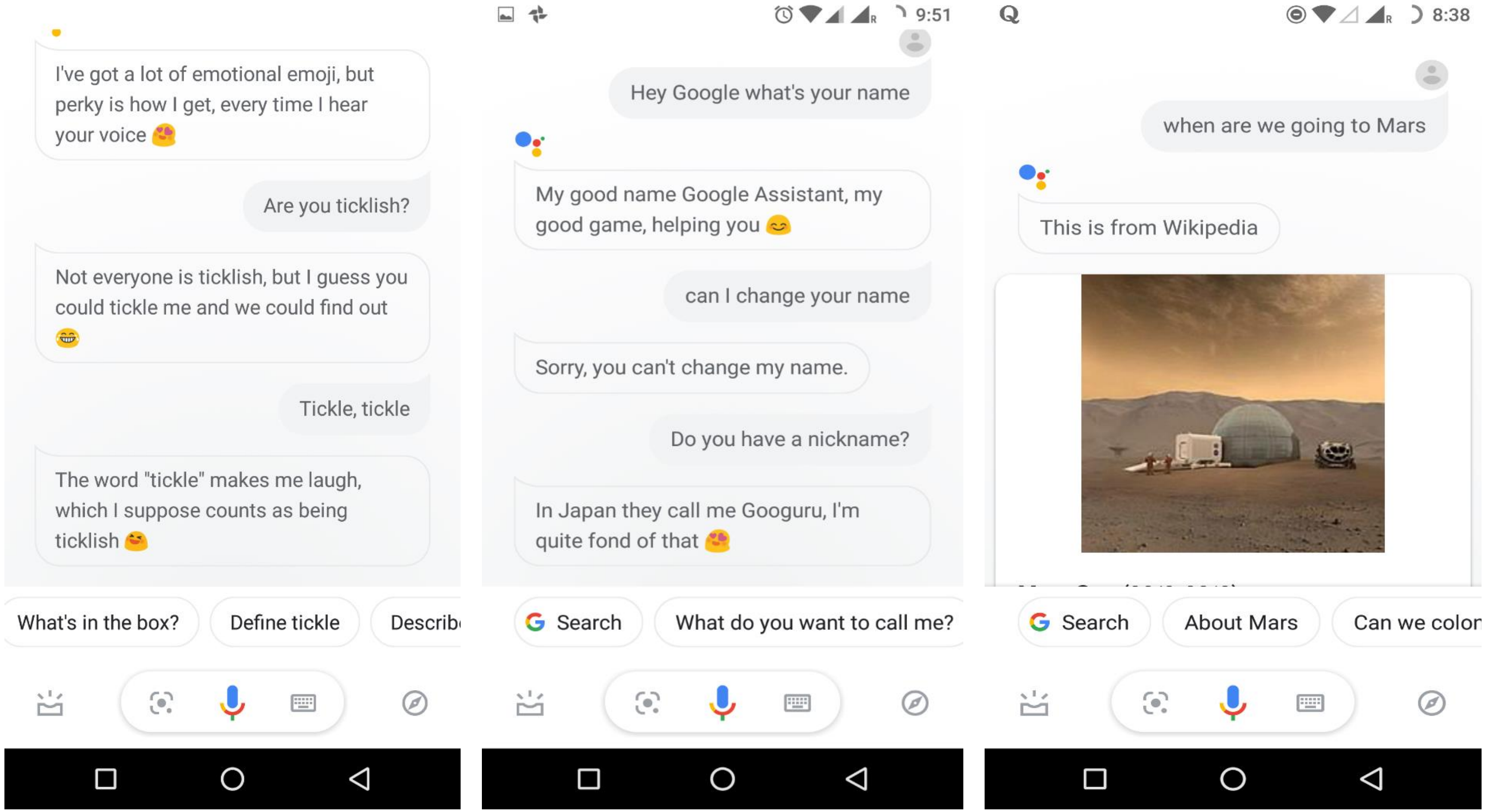
- Personal Assistants like Google Assistant, Apple’s Siri, Amazon Alexa, etc. are the culmination of open-ended and goal oriented dialog systems.
- The domain of input is not restricted to certain use cases only.
- The user can enter anything he/she feels to.

Goal-Driven/Task-Driven



Social Dialogues (Our Focus)

Open Ended Dialogues comes into the category of social dialogues, as evident they can be anything, there is no particular domains to which these dialogues belong. And can sometimes be difficult to deal with.



Why do we need Conversational Agents?

- Convenient to just say or write what we want a machine to do.
- Exploiting all features of a certain machine which can be cumbersome to do if done manually.
- Need a way to interact with machines similar to how humans interacts with one another.
- Exciting results from other Natural Language Generation tasks.
- Is Fun…

Problems to Open-Domain Conversational Agents

- Repetiveness: The inability of model to give diverse set of output given same set of inputs.
- Specificity: The response to be related to previous context of the conversation.
- Knowledge(or any other trait): The ability for the dialog models to have general common knowledge or enough knowledge to support the conversation.
- Consistency: The ability of the conversational agent to not to contradict itself during the course of the conversation.

```
Human> everything is not good today
Bot: not again
Human> everything is not good today
Bot: not again
Human> quit
```

Repetition

```
Human> what do you work ?
Bot: i ain t a dream .
Human> what is it for ?
Bot: nothing . not bad .
Human> quit
```

Specificity and Response-Relatedness

These are some of the examples from our early sequence to sequence model.

Making Better Models

- We compared two classes of models:
  - RNN based Sequence to Sequence Architecture
  - Transformer based Sequence to Sequence Architecture

- Transformer based model performed almost 100x better than the RNN based model despite having almost half the parameter than the RNN based.

Model	Unigram F1	Perplexity
RNN based seq2seq	0.0887	38034340.2348
Transformer based seq2seq	0.5943	114691.363

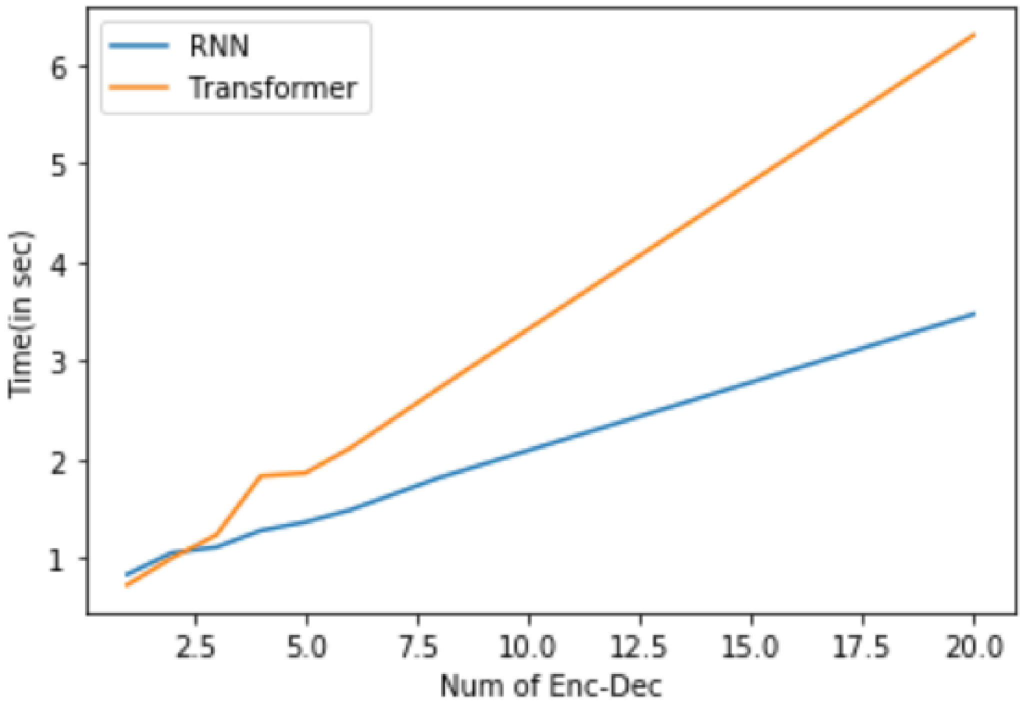
Although we achieved great success while evaluating through automatic metric. The conversational experience difference was not so subtle.

Efficiency Comparison

- We need efficient models that allow parallelisation for scaling with large texts.
- RNN based models do not have the ability to parallelise as the computation of current hidden state depends on the previous hidden states.
- Transformer allows suitable parallelisation.

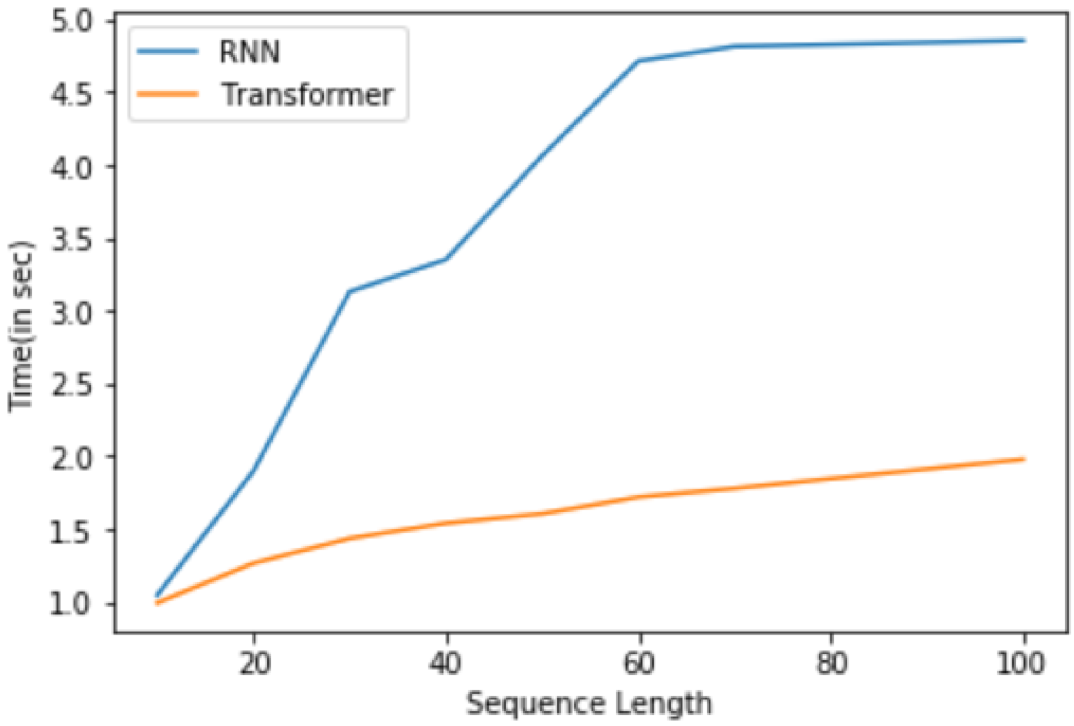
Keeping Sentence Length Constant

For small sequences RNN based architecture runs much faster than the Transformer model.



Increasing Sequence Length

The Transformer is really efficient when it comes to long large sequences. Which is what we needed for large corpuses.



Different Decoding Algorithms

- We also experimented with different decoding algorithms.
- Decoding is the way by which produce text from a trained Natural Language Generation model.
- We experimented with greedy decoding, beam search, Top-k and Top-p sampling(Nucleus Sampling)[Released just in time].
- The top-p, top-k sampling decoding algorithms provides better diversity in responses.

```
Human: hello
Machine: hi .
Human: what are you doing ?
Machine: i m sorry .
Human: which is you favourite song
Machine: it s why you been there .
Human: okay
Machine: you ll get it .
Human: whom do you live with ?
Machine: the store .
Human: where do you live ?
Machine: right here .
Human: who do you work for >
Machine: for seven will do you ?
Human: who do you work for ?
Machine: for a stupid friend .
Human: oh
Machine: what ?
Human: quit
```

With random sampling, top-p, top-k sampling decoding algorithms we observed diverse set of outputs for same set of inputs and a much better conversational experience.

Including Context

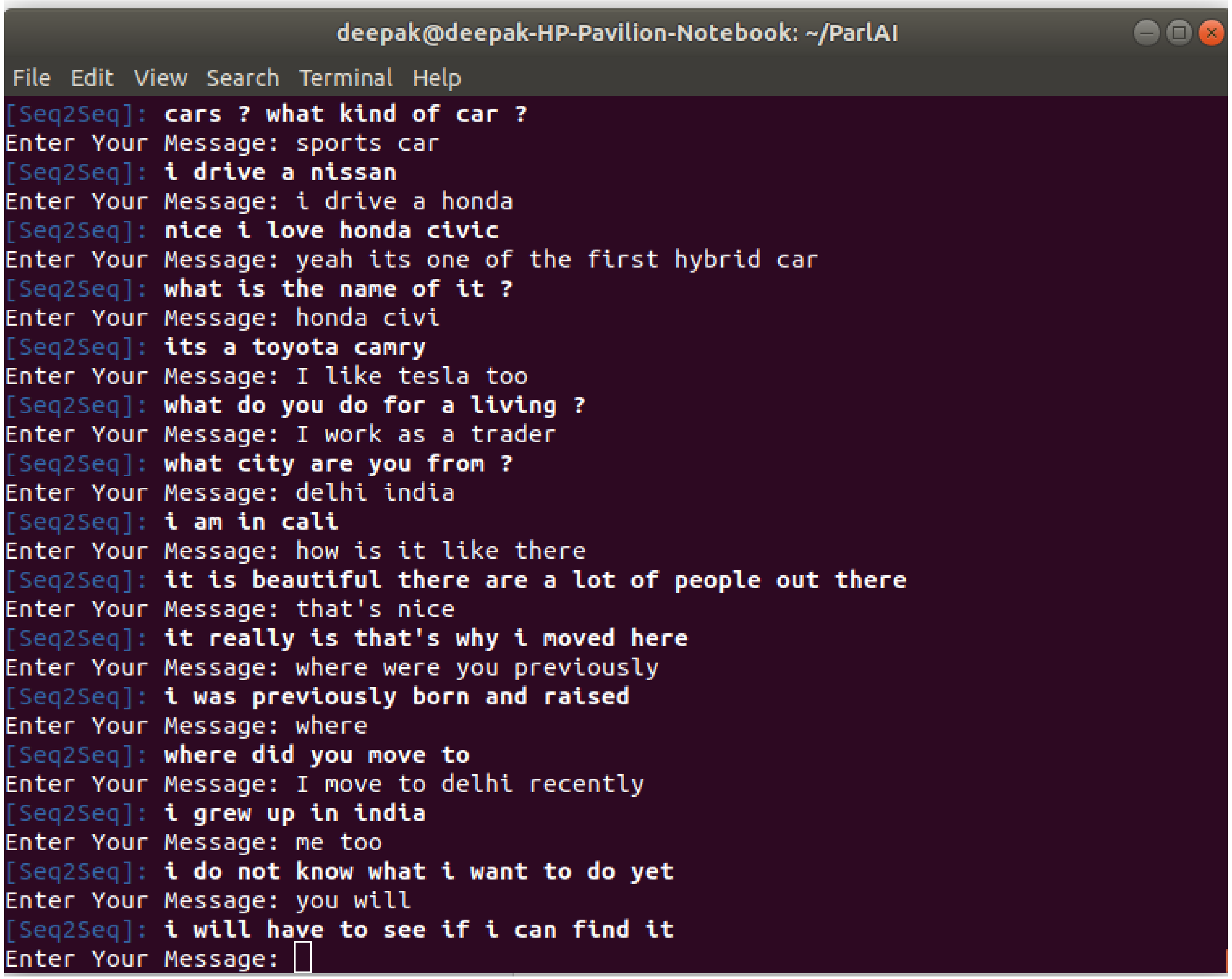
- We studied two models to include the previous history of dialogues as well to generate the next response.
- HRED(Hierarchical Encoder Decoder)
- General Methodology
- We also studied how to include traits like explicit knowledge into our model.

Topic:	Shower (*unseen*)
Human:	So I've been looking for a better showerhead with more water pressure for my shower.
Model:	I love pressure-versed water at nearly constant temperature to a shower
Human:	That sounds pretty great! Love those rainshower showers!
Model:	the ancient greeks were the first people to have showers.
Human:	Well I knew they were good for something.
Model:	the original showers were neither indoor structures nor man made but were common.
Human:	Ah that sounds intimidating!
Model:	it is required manual transport of both fresh and waste water.

Knowledge based conversational dialog model. Snippet is taken from Dinan et al 2019.

Adding Control

- Adding explicit control to output responses. These control offers us to make the model's response more rich in certain areas like question-asking, specificity.
- We studied two class of methods for addition of control in dialog systems:
  - Weighted Decoding(WD): Different version of decoding algorithms.(During Testing)
  - Conditional Training(CT): Augmenting during training.



Adding control greatly improves the quality of conversation. With correct amount of question asking parameter, specificity-control, we can have good conversation with these systems.

Conclusion

- Use Transformer or Transformer based(Transformer XL,etc) models for Natural Language Generation tasks. No reason not to use them.
- Use of Top-p sampling algorithm greatly improves diversity.
- Adding Explicit knowledge does improves the system but takes up a lot of memory.
- Adding balanced control to our system is beneficial.

Future Work

- Reinforcement learning based open-ended conversational agents.
- Pre-trained-Language Model fine tuning them for conversational task.
- Large parametric models to get hold of complex structures within the data.
- Better models and algorithms to capture the innate structures.

Contribution Breakdown

- During the course of the project, we all wore different masks to successfully complete the project. We all did reading and learning part of the project.
- **Deepak Agrawal:** Ideas to be implemented, Conversion to Latex.
- **Nitanshi Mahajan:** Writing thesis, Presentation.
- **Deepak Goyal:** Implementation of Ideas, Managing things.
- Also, we would like to thank **Hitesh Kumar** for helping with some errors and implementation particularly helpful in chapter 4,5.

Questions

Thank You