PRESENTION ANSWERS

1. Why did you choose this dataset, and how did you preprocess it?

I decided to create my own dataset because I couldn't find one that really captured the kind of conversations I wanted which was empathetic, short, and suitable for people dealing with dementia. I called it *empathy-dementia*. It's small, about 170 samples, but each one is carefully annotated with things like emotion, intent, tags, and whether it's dementia relevant. That way the model isn't just learning random responses, but ones that make sense in caregiving situations.

For preprocessing, I kept it very simple but consistent. I used text prefixes like *"emotion:"* in English or *"émotion:"* in French to guide the model, then tokenized the data to a max length. I dropped metadata fields that weren't needed for training. The idea was to keep the inputs structured enough for the model to learn, without making the pipeline too heavy.

2. Why did you choose this modelling approach?

I went with **T5-base** as my main model. The reason is that T5 is good at conditional text generation—you give it a prefix or instruction, and it knows how to respond. That fits perfectly here because I can steer it with things like "emotion" or "intent," and it gives me controlled, empathetic replies.

I did look at other options. A classification + template approach would have been easier, but it felt too robotic, not natural enough. And generic chat models like DialoGPT or even larger LLMs can sound fluent but they're not tuned to the sensitive context of dementia care—they often drift or give responses that aren't appropriate. So, for me, fine-tuning T5 gave the right balance of control and empathy.

3. What challenges did you face with the dataset and feature engineering?

The biggest challenge was the size of the dataset. With only about 170 examples, it's easy for a model to overfit or to miss rare cases. I had to keep the training conservative—only a few epochs, small batch sizes, and constant validation checks.

Another challenge was working with bilingual data, since I included English and French. To handle that, I used consistent prefixes so the model wouldn't get confused, and I kept preprocessing light so the model could generalize.

And finally, integration was a challenge. Since I also built a live demo pipeline with Whisper for speech-to-text, SpeechT5 for text-to-speech, and emotion detection, I had to make all those parts talk to each other cleanly. I solved that by exposing everything through FastAPI endpoints, so each model is just a service I can call.

4. How do your results compare with existing work, and what's new about your approach?

What I noticed is that the model learned well from this dataset. The training and validation losses both dropped steadily, which means it wasn't just memorizing but generalizing. By the last epoch, the validation loss was very low, so I kept that checkpoint for deployment.

Compared to generic chatbots, my model doesn't just give random or casual replies. It consistently responds in a short, calming, and empathetic way, which is exactly what's needed for someone with dementia. That's where the novelty comes in, I built a small but domain-specific dataset, fine-tuned a model on it, and deployed it as a working empathy generator. And I tied it into a live conversational pipeline with ASR, emotion detection, and TTS. That combination of empathy plus end-to-end deployment is new.

5. If you had more time or resources, what would you do differently?

If I had more time, the first thing I'd do is expand the dataset, get more samples, balance the different emotions and intents, and maybe add local languages like Yoruba or Igbo so it's useful beyond just English and French.

Second, I'd work on multi-turn memory and reminders, because in dementia care it's not just about a single reply, it's about continuity and remembering what was said, reminding the person about important things, and giving cognitive exercises.

Third, I'd add stronger evaluation. Right now, I mostly tracked training and validation loss, but with more time I'd bring in human evaluations and automatic text metrics to really measure empathy and appropriateness.

Explanations for Evaluation/Validation/Training graphs

Figure 4.11 — System Evaluation Metrics Testing

This figure summarizes end-to-end behavior of the empathy generator on held-out data. The key metric we tracked during development was **cross-entropy loss**, which reflects how confidently the model assigns probability to the reference empathetic reply. Lower is better. As seen here (and corroborated by the model's training table), the **validation loss steadily decreased across epochs**, indicating the model learned stable mappings from dementia-relevant inputs (often with emotional cues) to brief, supportive responses without obvious overfitting. Qualitatively, generations align with emotion/intent labels from the dataset (e.g., friendly, caregiving) and remain concise and calming the important for the target users.

Figure 4.12 — System Validation Testing graphs

This graph shows the **validation curve** used for model selection. We evaluated at the end of each epoch and **kept the checkpoint with the best validation loss**. A smooth, downward trend with no late epoch rebound suggests the chosen schedule (4 epochs, small batches) was conservative enough for this small corpus. For future work, we'll complement loss with **human ratings** (clarity, empathy, appropriateness) and automatic text metrics to triangulate quality.

Figure 4.13 — Model Training graphs

These training curves track **training vs. validation loss** across epochs. The **parallel decrease** shows the model is fitting the task while maintaining generalization. The final epoch exhibits the **lowest validation loss (~0.038)** in the model card's table, which is the checkpoint we use in the **Empathy API** Space. Decode settings (beam search, early stopping) further reduce response variance at inference to induce the AI response faster while retaining the model functionalities. Hugging Face+1