WASP SE Assignment

Haotian Qi

August 31, 2025

1 Introduction

My research is focused on how people take turns in daily conversations and how to build systems that can model this process and deploy it to robots. In everyday conversations, people do not talk randomly over each other. Instead, there is a coordinated system that helps each speaker take turns to be the active speaker. This coordination can be guided by audio cues from the current speaker's speech, or by visual cues such as facial expression and gaze. In different environments, the turn-taking process also changes. For example, on the phone, people rely only on audio cues and have less access to short feedback signals like "hmm" or "yeah." In online meetings, where the speaker and listener both lack visual feedback, speakers often leave longer pauses before the next turn begins. For this research, the focus is on multiparty turn-taking in face-to-face settings, with the goal of building conversational systems for robots.

Turn-taking not only occurs after one person stops talking. It is an incremental prediction process, where participants anticipate when another person will finish and prepare their response in advance. Human conversation depends on these short predictions and rapid transitions. Current conversational agents, however, do not use such mechanisms. Most systems rely on silence detection and respond only after a fixed pause. For example, many applications treat a 200 ms silence as the signal to take a turn. This often causes problems when the user is still in the middle of constructing a sentence. The result is an interaction that resembles text-based messaging, where each side delivers a block of speech in sequence rather than overlapping, signaling, or adjusting in real time. As a result, the flow is slow and unlike natural human dialogue.

2 Lecture Principles

2.1 Input Slicing Testing

Ablation study is similar to input slicing testing. The task is to investigate which parts of the model contribute to its predictions. Instead of relying only on accuracy, F1 score, or balanced accuracy, we create multiple subsets of test data that focus on specific properties for more robust evaluation.

For multimodal multiparty turn-taking modeling, we need to test multiple aspects of performance. For example:

Number of Speakers: Instead of combining all ranges of speakers in the test data, we compute metrics on subsets grouped by the number of speakers. For two speakers, the task is simpler because only one potential speaker can take the turn, making the prediction binary.

Interruption Type: In casual conversations, people often talk over each other and initiate turns more freely. In formal settings like interviews or podcasts, listeners wait longer or pause before taking a turn. Evaluating on these different types helps measure how the model handles interruptions.

Acoustic Prosody: We can manipulate the audio using filters such as lowpass, highpass, reducing energy, flattening F1, or adding noise. F1 here refers to the first formant frequency in speech,

which relates to vowel quality. Energy refers to the signal amplitude, which correlates with loudness or emphasis. Lowpass and highpass filters remove certain frequency ranges, allowing us to test which prosodic cues the model relies on. These manipulations reveal weak points and quantify the contribution of prosody to multimodal modeling.

Silence vs Anticipation: Instead of checking predictions only after a speaker pauses, we also evaluate before the turn occurs. This measures how often the model can anticipate a shift before the active speaker finishes speaking.

2.2 Cognitive bias

In multimodal machine learning, feature selection is one of the easiest ways to fall into the pitfall of cognitive bias. For my task of turn-taking, there are many hidden visual features that might be crucial for prediction, such as emotions, blinking, eye gaze, and facial expression. But it is easy to simply pick already existing pre-trained encoders for other tasks. For instance, face embedding or lip embedding that are trained on another task can then be adapted for use in our model. In my current work, we train a visual encoder from scratch that focuses on turn-taking prediction, which allows it to capture the things that we want. In general, we do not feed second-level features and tell the model to learn from them. We feed the raw image, exactly what humans see, and let the model itself learn which features might be useful.

3 Guest-Lecture Principles

3.1 Human Aspect

In my research, we put our model onto the robot and monitored the interactions between humans and the robot. One of the key aspects is how much the human trusts the AI. For instance, in one experiment our colleagues scanned brain signals while participants had a conversation with a real human and with an AI. Within this experiment, both the human and the AI used the same digital 3D model to answer and interact. Some results show that by the nature of humans believing they are talking to a human, the way they speak and handle information changes a lot. In some cases, when they were talking to an AI but believed they were talking to a human, they became even easier on the target. This is important for my research since my turn-taking prediction will help monitor the real-time interactions between the robot and the people.

4 Data Scientists vs. Software Engineers

I do agree that there are essential differences between data scientists and software engineers. Data scientists focus mainly on model performance and data input. Software engineers focus on user experience and user feedback. With the example of audio transcription, the expertise of the data scientist will focus more on what type of data the user sends and receives, and how accurate it is for different types of audio and accents involved. The software engineer will use the out-of-the-box model as an anchor point and build the whole system around it, like UI, interface, and other components to maximize the user experience and to deploy it to the vast majority of users. Additionally, they build safeguards in case the model fails in certain circumstances.

I believe the roles will evolve and specialize further depending on the complexity of the data. For instance, prompt engineering was a trending job position that only focused on how to modify the input for large language models. Also, there are positions like ML engineering where some of them focus a lot on how to combine different components and set them up in a way to reduce the latency of models when deploying them to the general public.

5 Paper Analysis

5.1 Conversation Agent QA test

The first paper I selected is: "Quality Assurance of Generative Dialog Models in an Evolving Conversational Agent Used for Swedish Language Practice [1]"

5.1.1 Core ideas and SE importance

The paper introduced a set of 38 requirements with 15 automated tests to evaluate the quality of generative dialog models (GDM). They focus on ensuring the conversation flows from the conversation agents through requirements, design, and automated test cases. They argue that QA testing for GDM is difficult because of natural biases and vague sentiment in conversations, as mentioned in previous sections. It is hard to justify the performance of a generative model by metrics alone, such as accuracy, F1-score, or training loss on the dataset. When these models are deployed and used by real people, a large portion of the true performance and feedback has to come from the human perspective and the feeling of using this product.

The motivation is due to their belief that language learning is crucial for migrants and that scalable AI personal language tutors can significantly improve the learning curve when the AI agents can respond 24/7. They also emphasize that GDM need to be correctly filtered for things like bias, toxicity, and misinformation across different cultures and contexts during learning. For things like some words or translation might be off if the AI does not adapt to the current new topic, which new word might bypass the generic toxic filter. They argue that good QA tests should be designed and tailored specifically for GDM, and they experiment with a conversation agent called Emily, a Swedish-language GDM, as a case study.

For engineering importance, QA designed for GDM is often missing and hard to transfer. For instance, large language models are often tuned for more general purpose by using reinforcement learning with human feedback, so a QA designed for a subtask should be tailored for the target demographic. Also, NLP models often require metrics beyond accuracy or F1 score. They often require measures of fluency, coherence, and confidence to ensure that users will not misinterpret or "gaslight" the GDM.

5.1.2 Relation to my research

My research focuses on turn-taking prediction for human-robot interactions, and the end goal is to deploy such models to change the dynamics of conversation. In this sense, both research areas share the field of human-robot interactions, and the QA tests in the paper also focus on how models are tested and quality-assured.

For instance, my model fundamentally allows the robot to actively initiate the conversation with the user by observing visual features like gaze, emotions, and expressions. Additionally, the agent uses these expressions to indicate when it is the user's turn to respond. Currently, there are no test cases or evaluation methods for our scenario since it involves a completely new model and usage.

5.1.3 Integration into a larger AI-intensive project

This paper focuses on a simple agent called Emily, but it could be improved and tested on a broader scale. First, we could modify and scale it to more languages. Additionally, these test cases could be integrated as a subset of QA tests for generic conversation agents, not just for language study. For instance, a large-scale, multi-language AI tutor could use this to perform iterative testing for each expert model to ensure quality is maintained.

For my research, I could use these test cases to ensure that my model does not create strange or uncomfortable gestures or expressions, making sure it aligns with our goals.

5.1.4 Adaptation of my research

The core value I captured from the paper is how to design test cases for downstream tasks. In the paper, they include every step of how they formalize the problems. In the first step, they build a table of requirements and assign points for each combination using three metrics: Value, Effort, and Novelty. This approach can be used to build a more structured and software-engineering-oriented framework.

In the test case categories, I could adapt the way they evaluate the personality traits of the GDM model. They group these cases with a focus on Toxicity, Nagging, and Stuttering. For the second stage, my model will focus more on expression generation, where the model gives out its own visual cues as feedback for the users. At this stage, our model is effectively a generative model that directly outputs feelings or cues for a task. Here, we need to ensure that the expressions are valid and comfortable for the users, since robots often fall into the uncanny valley, which makes people feel uneasy. Additionally, we need to make sure we evaluate the expressions and the way the agent speaks against the toxicity, nagging, and stuttering tasks.

5.2 Understanding Quantization

The second paper I select is: "Towards Understanding Model Quantization for Reliable Deep Neural Network Deployment[2]"

5.2.1 Core ideas and SE importance

Quantization is a common method that allows us to compress a model into a smaller one by changing the floating-point precision. The idea is to enable running a model on much smaller devices that often do not have powerful GPUs, such as mobile devices or Internet of Things systems.

The main focus of this paper is the prediction disagreement that occurs after performing quantization. This is defined as the case where the same input produces a different output in the compressed model compared to the full-precision version. They argue that accuracy change is a poor choice of metric and can be quite misleading for different types of behavior. For example, they show a case of compressing a pre-trained model where both versions yield the same accuracy, but the compressed model produces 216 prediction disagreements. The paper suggests that we should also monitor the confidence score for the top class. By checking this score, the model can recognize when it is uncertain and act more cautiously instead of making direct predictions.

The paper shows that trying to fix the model by retraining on the compressed version is not effective. When one disagreement is fixed, a new one tends to emerge. They claim that the best method is Quantization-Aware training, which prepares the compressed model during the training process itself. A similar approach can be seen in knowledge distillation, where using a distillation loss allows the model not only to train for better full-scale performance but also to create greater separability when the model is scaled down.

5.2.2 Relation to my research

My own work focuses on getting robots to have natural conversations with humans, which means they need to see and hear things in real time on the robot. However, since I am doing multimodal modeling, video processing often has significant computational costs. Compared to text, which is a sequence in one dimension, and images, which have two dimensions (width and height), video requires an additional dimension—time. Therefore, for the ability to deploy my model, quantization is the most

straightforward solution, and this paper dives deep into the "disagreement" and "perceptual mistakes" that might occur.

5.2.3 Integration into a larger AI-intensive project

Assume we have a real project that requires the robot to understand people's facial expressions and tone of voice in order to be a good companion. For privacy and latency reasons, all the AI has to run locally on the robot itself. This requires us to compress the model, potentially using quantization.

By utilizing the papers, we could apply quantization-aware training methods to make the model more robust and less likely to make mistakes after compression. While running, the robot would always check the confidence score to evaluate how certain it is across different input modalities.

The final goal of my WASP project is to build a conversational robot model that can detect visual cues. For example, when the user seems unsure about something, or when the robot senses that the person wants to speak in the middle of its own speech, it would pause and ask, "Sorry, did you want to say something?" This differs from traditional conversational robots, where each party simply takes turns and the user has to wait until the robot finishes speaking. Our system would appear smarter. Therefore, the proper timing of pauses or decisions is crucial, and it should not drift into the wrong moments after compression.

5.2.4 Adaptation of my research

This paper gives me insight into thinking about quantization in advance, in terms of considering compressed model performance early on. I will also make the next version of my conversation model incorporate confidence scores as part of the decision-making process. For example, when the robot is listening to a conversation or a stream of input, the previous version of the model only cared about which class label prediction was the top-1 and which action it should take to improve the conversation. Now, the model will also monitor the difference between the first and second class probabilities. This enables the model to "think" a little more and wait for the proper moment to make a decision.

Alternatively, we could use this uncertainty as part of the class label, allowing the model to actively ask clarifying questions like: "Could you repeat that?" or "Sorry, I didn't understand the last part." This creates a new type of zero-shot tuning in the downstream task, by simply comparing two types of metrics: probability vs. confidence.

In the long term, I want to change how my model is built altogether. In the current scope, I only have a model and a potential brand of robot where it needs to be deployed. This gives me the suggestion of taking compressed model performance into consideration from the start. This approach shifts the goal from only valuing performance on a test dataset to focusing on which parts of the model's capability are most vulnerable after we perform quantization.

6 Research Ethics & Synthesis Reflection

6.1 Search and Screening Process

I first use the official cain conference website which stored the paper in accepted paper sections. But then I realize there is not enough paper or more to say they didnt list all of the paper. So I use the IEEE portal to check all the papers there is on the CAIN conference.

For the purpose of relate to my own research, I scan through all the papers from 2025 to 2022 and check whether anything related to things like conversational AI. If there is no close related, I search for another keyward for things like new metrics for validation. The ideas is that which papers that could potentially get used and cited by my own work in the future. I then quick read the abstract, method and conclusion to determine whether it is fit to my own research.

6.2 Pitfalls and Mitigations

Because the way I search and screening for papers which really related to my own project, it is easier for me to check whether there is misleading title or abstract. Assume, I find the paper that had those issue, I would expand the range of selection base on the level of understanding of I had by just looking at the title and abstract.

6.3 Ethical Considerations

For not taking any copy from LLMs or sources, I build the structure of the assignment close to my own research field as much as possible. By thinking using my own research knowledge as fundation to do the screening process, I would be much easier to spot the part I deemed is useful. Also, the text will be spill out with really rough draft in a really casual talking manner, and dumpping my thoughts. Then I go back and finilize and finish the whole process and fix grammar.

References

[1] Markus Borg, Johan Bengtsson, Harald Osterling, Alexander Hagelborn, Isabella Gagner, and Piotr Tomaszewski. Quality Assurance of Generative Dialog Models in an Evolving Conversational Agent Used for Swedish Language Practice . In *2022 IEEE/ACM 1st International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pages 22–32, Los Alamitos, CA, USA, May 2022. IEEE Computer Society.

[2] Qiang Hu, Yuejun Guo, Maxime Cordy, Xiaofei Xie, Wei Ma, Mike Papadakis, and Yves Le Traon. Towards understanding model quantization for reliable deep neural network deployment. In *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pages 56–67, 2023.