Meeting Transcript: RPA Process Automation Discussion

Transcript:

Mark - 10:00:12: Hey everyone! How are you doing today? It's quite chilly outside, isn't it?

Julie - 10:00:18: Good morning! Yes, it's freezing. I had to scrape ice off my car windshield this morning.

John - 10:00:27: Morning all. Actually, it's not too bad compared to last week. At least the sun is out today.

Mark - 10:00:41: True, that makes a difference. Well, before we dive into details, I think we should introduce ourselves since this is our first meeting together. I'm Mark from the development team, and I'll be your main point of contact for this project. I help translate business needs into technical requirements.

John - 10:01:03: Nice to meet you both. I'm John from the Operations department. I'm looking for help automating our meeting summary process. Currently, I spend hours each week creating and distributing meeting summaries from our Teams calls, and it's taking up too much of my time.

Julie - 10:01:22: Great to meet you, John and Mark. I'm Julie, and I'm an RPA developer. My job will be to build the automation solution based on your requirements. I specialize in Python-based RPA solutions, so I'm excited to hear more about this process and how we can automate it effectively.

Mark - 10:01:41: Perfect! Now that we know each other, John, could you walk us through your current process for creating meeting summaries? The more details you can provide, the better.

John - 10:01:55: Sure thing. So the process starts when I receive an email from Microsoft Teams with the meeting transcript attached. These emails come automatically after each meeting where recording and transcription were enabled. I typically get 5-10 of these each week.

Julie - 10:02:15: And what format does the transcript come in? Is it a text file, Word document, or something else?

John - 10:02:23: It comes as a .vtt file attachment in the email. Sometimes there's also a recording link, but I only need the transcript part.

Mark - 10:02:36: Interesting. And what do you do with this transcript file next?

John - 10:02:44: I open it, read through the entire thing, and then create a summary document. I try to identify the key discussion points, any decisions made, and action items assigned to people. It's quite time-consuming because some meetings are over an hour long.

Julie - 10:03:02: What does your final summary look like? Is there a specific format or template you follow?

John - 10:03:11: Yes, I have a standard format. The summary has four main sections: Meeting Details (date, time, participants), Key Discussion Points, Decisions Made, and Action Items with owner and deadline. I create this in a Word document with our company template.

Mark - 10:03:31: And how do you distribute this summary once it's complete?

John - 10:03:38: I email it to all meeting participants. I have to copy the participant list from the original Teams email and paste it into a new email with the summary attached.

Julie - 10:03:52: This sounds like a good candidate for automation. Let me make sure I understand the full process: First, you receive the transcript by email, then you manually extract important information, create a structured summary document, and finally email it to all participants. Is that correct?

John - 10:04:13: Exactly. And if we could automate this, it would save me several hours each week.

Julie - 10:04:22: What about the identification of key points, decisions, and action items? Are there specific keywords or patterns you look for when creating these summaries?

John - 10:04:37: Good question. I just read through the transcript and look for what we agreed to do. For key discussion points, it's trickier - I basically summarize the main topics discussed.

Mark - 10:05:01: Would you be comfortable with an AI-assisted approach for the summary creation part? We could use large language models to identify those elements.

John - 10:05:14: Absolutely, I'm open to that. As long as I can review the summary before it goes out, at least initially to make sure it's capturing everything correctly.

Julie - 10:05:27: That makes sense. Do you have any specific requirements for the email subject line when you send out these summaries?

John - 10:05:38: Yes, I always use the format "Meeting Summary - [Original Meeting Title] - [Date]". It helps people find them later.

Mark - 10:05:49: Is there anything else about your current process that we should know?

John - 10:06:02: One thing to note is that sometimes I need to clean up the transcript text a bit before processing. The automatic transcription isn't always perfect with technical terms or when multiple people speak at once.

Julie - 10:06:18: That's helpful to know. I think I have enough information to start designing a solution. To summarize what we'd automate: monitoring your inbox for Teams transcript emails, extracting and processing the transcript, creating a structured summary document with the four sections you mentioned, and then emailing that summary to all original participants with the correct subject line format.

John - 10:06:44: That sounds perfect. And if I could have a quick review step before the email goes out, even better.

Mark - 10:06:53: Great! Julie, do you foresee any technical challenges with this automation?

Julie - 10:07:04: The email monitoring and sending parts are straightforward. For the core summary generation, I think we should leverage Large Language Models rather than traditional pattern matching. Models like GPT-4 or Claude are excellent at understanding context and extracting

relevant information from transcripts. We'll need to set up some initial training with John's historical summaries to get it right, if those are available.

John - 10:07:26: I can provide at least 20 examples of previous transcripts and the summaries I created from them. Would that help?

Julie - 10:07:35: That would be excellent. Those examples will help us fine-tune our prompts to the LLMs to generate summaries in your preferred style and format. We'd use them as few-shot examples to guide the AI.

Mark - 10:07:48: I've heard a lot about these LLMs. Would we be using an API to access them?

Julie - 10:07:56: Yes, we'd integrate with either OpenAI's API for GPT models or Anthropic's API for Claude. Both offer excellent capabilities for this type of text processing and summarization. The benefit is that these models understand natural language nuances much better than rule-based approaches.

John - 10:08:16: Will this approach be more accurate than traditional methods?

Julie - 10:08:22: Significantly so. Traditional methods like regex patterns would struggle with the contextual understanding needed for good summaries. LLMs can understand discussions even when they don't follow predictable patterns or use specific keywords. They're also much better at producing natural-sounding summaries that capture the essence of the conversation.

Mark - 10:08:44: Are there any drawbacks to using LLMs that we should be aware of?

Julie - 10:08:52: Good question. There are a few considerations. First, there's a cost associated with API calls to these models, though it's generally reasonable for this volume of work. Second, we need to be mindful of data privacy since we'd be sending transcript data to external APIs. And third, there can occasionally be latency issues depending on the API load. We'd need to implement proper error handling and retries.

John - 10:09:22: The privacy concern is important. Do these services store our data?

Julie - 10:09:30: Most providers have options to not store your data. For example, OpenAI and Anthropic both offer data privacy options where your inputs aren't used for model training and aren't retained beyond the processing time. We'd definitely configure those options for this application.

Mark - 10:09:48: That sounds promising. Let's dive a bit deeper into how you'd implement this solution using LLMs. What would the workflow look like?

Julie - 10:10:01: The workflow would be relatively straightforward. First, we'd monitor your inbox for Teams transcript emails using Python's email libraries. When a new transcript arrives, we'd extract and parse the .vtt file to convert it into clean text. Then we'd craft a prompt for the LLM that explains the task and includes examples of how you typically create summaries.

Julie - 10:10:28: The prompt would instruct the model to extract meeting details, identify key discussion points, list decisions made, and compile action items with owners and deadlines. We'd

specifically use Claude for this part, as it's particularly good at following structured instructions like this.

John - 10:10:47: How would you ensure it captures all the action items correctly? That's the most critical part for our team.

Julie - 10:10:56: Great point. For action items specifically, we'd use a two-stage approach. First, we'd have the LLM identify potential action items throughout the transcript. Then, we'd run a second pass specifically focused on these candidates to extract the responsible person, the task details, and any mentioned deadlines. GPT-4 is particularly strong at this kind of structured extraction.

Mark - 10:11:22: Could we also have it verify the action items by checking if they follow a logical pattern? Sometimes people mention tasks that aren't actually assigned to anyone.

Julie - 10:11:34: Absolutely. We could add validation logic that has the model check each action item to confirm it has a clear owner and description. If there's ambiguity, we could flag those items for human review. The benefit of using LLMs is that they can apply this kind of reasoning that would be very difficult with traditional programming approaches.

John - 10:11:57: This sounds really promising. Would it be able to handle technical jargon and acronyms specific to our industry?

Julie - 10:12:07: Both GPT-4 and Claude have extensive knowledge of most industry terminology, but we can further enhance this by providing domain-specific context in the prompt. We could include a glossary of your company's unique acronyms and technical terms that the model should recognize. For very specialized terms, we might fine-tune a smaller model specifically on your domain language if needed.

Mark - 10:12:32: How would the review process work before sending the summary?

Julie - 10:12:39: We'd build a simple web interface where you can view the LLM-generated summary side by side with key sections of the original transcript. You'd be able to edit any part of the summary directly in the interface, and once you're satisfied, click a button to approve and send it. Over time, we could track your edits to further refine the prompts and improve accuracy.

John - 10:13:04: That sounds ideal. What about handling multiple languages?

Julie - 10:13:11: That's actually a great advantage of using LLMs. Both GPT-4 and Claude are multilingual models that can understand and generate content in numerous languages. If you have meetings in Spanish or other languages, the models can process those transcripts and generate summaries in the same language or even translate them to English if preferred.

Mark - 10:13:35: What about the document generation? Would the LLM create the actual Word document?

Julie - 10:13:42: The LLM would generate the structured content, which our system would then format into your Word template using the python-docx library. We'd map the AI-generated sections to the appropriate parts of your template, maintaining all your corporate formatting, headers, and styling.

John - 10:14:00: How quickly would this process run? Currently, summarizing an hour-long meeting can take me 30-45 minutes.

Julie - 10:14:10: The entire process from receiving the email to generating a summary ready for review would typically take less than a minute. LLM processing is quite fast, especially for text of this length. The longest part might be the initial email detection and extraction, but even that should be quick. You'd receive a notification as soon as a summary is ready for your review.

Mark - 10:14:35: What about security beyond the data retention concerns we discussed earlier?

Julie - 10:14:43: We'd implement several security measures. All communications with the LLM APIs would be encrypted using HTTPS. Your email credentials would be securely stored using environment variables or a secrets manager, never in the codebase. The web interface would require authentication, and we'd implement session timeouts. Additionally, we'd ensure that the temporary files created during processing are securely deleted after use.

John - 10:15:12: This sounds very comprehensive. What's the timeline for implementing something like this?

Julie - 10:15:20: Given that we're using LLMs rather than building custom NLP models, we can move quite quickly. I'd estimate about three weeks for a fully functional prototype. The first week would focus on setting up the email monitoring and LLM integration. The second week would be for building the web interface and document generation. And the third week would be for testing, refinement, and security hardening.

Mark - 10:15:48: That timeline works well with our current schedule. Would there be any ongoing maintenance costs or considerations?

Julie - 10:15:58: The main ongoing cost would be the LLM API usage, which would depend on the volume and length of transcripts. For 5-10 meetings per week, I'd estimate $50-100 per month in API costs. There's also the hosting cost for the web application, which would be minimal. In terms of maintenance, we'd need to occasionally update the integration if the LLM providers change their APIs, but that's typically infrequent.

John - 10:16:28: That sounds reasonable. Would it be possible to add features in the future, like automatic distribution of action items to team members' task lists?

Julie - 10:16:40: Absolutely! That's a great extension idea. Once we have the LLM reliably extracting action items, we could integrate with task management systems like Asana, Trello, or Microsoft Planner to automatically create tasks assigned to the respective team members. The LLM makes this kind of extension much easier because it can structure the data so cleanly.

Mark - 10:17:02: We should also discuss how we'll measure the success of this implementation.

John - 10:17:09: For me, the key metrics would be time saved and accuracy of the summaries, especially the action items.

Julie - 10:17:18: We can definitely track those. We'd measure how long it takes from email receipt to approved summary as the time metric. For accuracy, we could track how many edits you make to

the LLM-generated summaries, with fewer edits indicating higher accuracy. We could also implement a simple rating system where you score each summary's quality from 1-5.

Mark - 10:17:42: Perfect. Any other questions or concerns, John?

John - 10:17:48: I think this covers everything I was wondering about. I'm excited to see it in action!

Mark - 10:17:56: Great! I'll send out a summary of today's discussion and next steps by tomorrow morning.

Julie - 10:18:05: I'll start working on the technical design right away, focusing on the LLM integration. I might reach out with some specific questions about your email system and the existing Word template.

John - 10:18:19: Sounds good. Thanks both of you for your time today!

Mark - 10:18:25: Thank you, John. Have a great day!

Julie - 10:18:30: Looking forward to working on this. Goodbye!