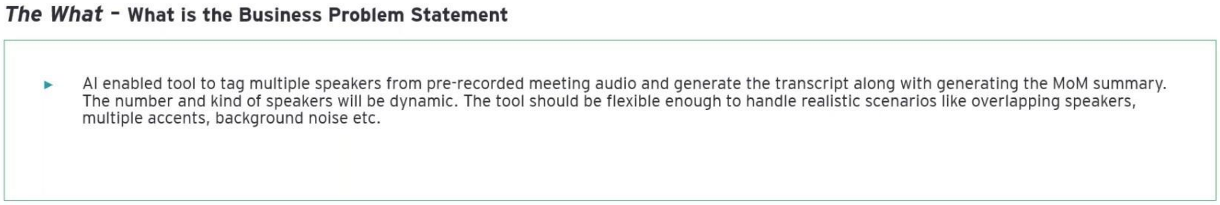
Theme 6: Intelligent Minutes of Meeting (MoM)

Introduction

Thank you for joining the Intelligent MoM webinar, being organized as part of the Radio – EY GDS Hackathon. In this session, we give you a quick overview about the theme, the expectations and answer some of the common questions that you may have.

Business overview

With everyone across the organization working remotely these days, virtual platforms are the new medium to connect and interact. A lot of content and data is discussed during these calls, making transcription a difficult task. There is a lot of manual effort required in preparation of a MoM – it should cover end-to-end content, identify/ list who has spoken on what topic/ subject, action points and other relevant details.



The What **– What is the Business Problem Statement**

- AI enabled tool to tag multiple speakers from pre-recorded meeting audio and generate the transcript along with generating the MoM summary. The number and kind of speakers will be dynamic. The tool should be flexible enough to handle realistic scenarios like overlapping speakers, multiple accents, background noise etc.

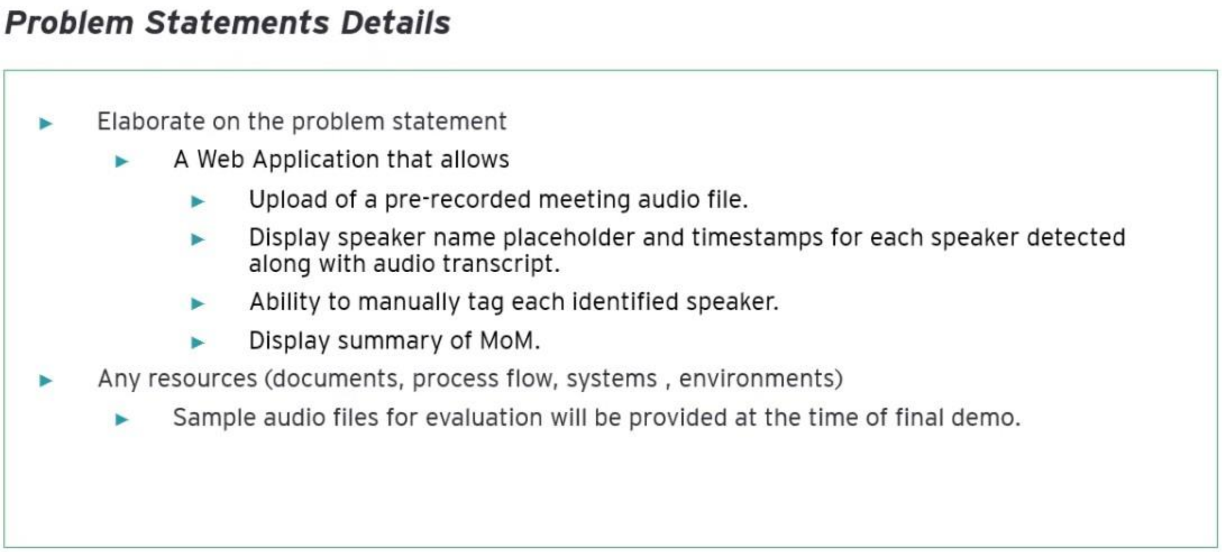
The input will be pre-recorded meeting audio/ video which will have multiple speakers, or participants. The input data need to be processed and transcribed to text. And, now, the system should enable identification of the speakers. If there are 10 speakers in that conversation, it should identify the statements, which are mentioned or discussed by the speaker 1, and then the speaker 2, etc. So, it should tag accordingly and that is just one aspect. Another aspect is to derive the action items from the discussion -who needs to do what and if possible, on what time.

Technical implementation

A couple of points to note here:

- We should not take the number of speakers or the specific speaker ahead of time, as a hard-coded feature. There should not be an assumption, like, only a specific set of users could be part of the meeting. The number of speakers and the participants will be dynamic, and the tool should be flexible enough to handle this. We are calling this out to prevent models being trained to work with only specific voices. So, if you assume that only those speakers in the training set will be the participants, the prediction will fail.
- The count can also be dynamically handled.
- The tool should also handle other realistic scenarios. Speakers might speak in an overlapping way, speakers might have multiple accents, and there might also be background noise, as well as other interruptions. So, ideally, those realistic scenarios should also be considered.

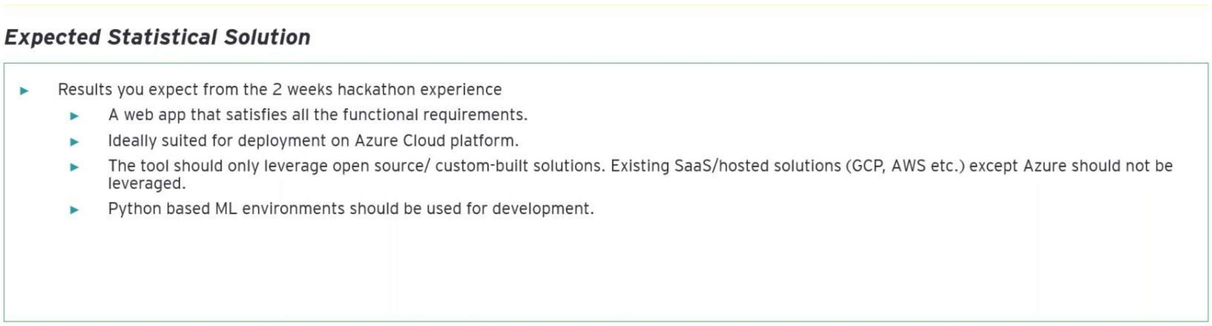
Implementation details



Problem Statements Details

- Elaborate on the problem statement
  - A Web Application that allows
    - Upload of a pre-recorded meeting audio file.
    - Display speaker name placeholder and timestamps for each speaker detected along with audio transcript.
    - Ability to manually tag each identified speaker.
    - Display summary of MoM.
- Any resources (documents, process flow, systems , environments)
  - Sample audio files for evaluation will be provided at the time of final demo.

We expect the solution to be a web application, where an audio file can be uploaded. The audio source will not be real time, it will be a pre-recorded audio file. The output shown by the web application would be the total number of speakers and the time stamps of what they speak, i.e. the transcript. So, the hackers have the flexibility to present this any way possible, in the user interface. Essentially, we need to generate who spoke what and the matching time stamps.



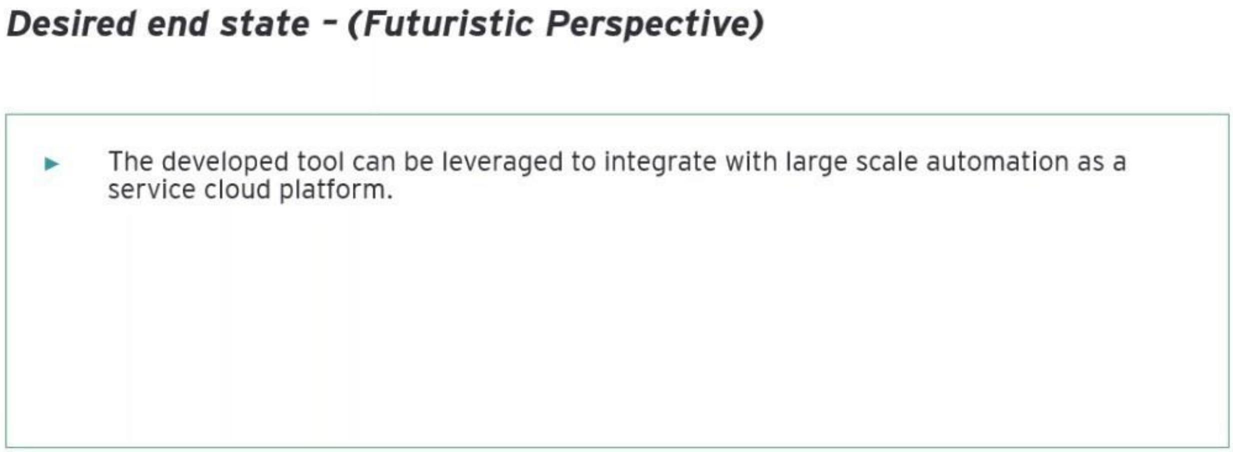
Expected Statistical Solution

- Results you expect from the 2 weeks hackathon experience
  - A web app that satisfies all the functional requirements.
  - Ideally suited for deployment on Azure Cloud platform.
  - The tool should only leverage open source/ custom-built solutions. Existing SaaS/hosted solutions (GCP, AWS etc.) except Azure should not be leveraged.
  - Python based ML environments should be used for development.

So, each speaker will initially be a placeholder, such as speaker 1, speaker 2, and so on. The end user should have the ability to manually tag each of the speakers. The artificial intelligence (AI) does not need to identify the speaker. A human operator, or the end user would know that speaker 1 is Hari and speaker 2 is someone else and so on. So, once they tag that, the transcripts just need to be updated. The speaker 1 needs to change to the tag name, and so on and so forth. And the same needs to be updated in the meeting minutes as well. For example, if the meeting minutes has a summary, that speaker 1 assigned a task to speaker 2 to be done on a particular day, that speaker 1 and speaker 2 will get updated based on the tagging.

Basically, that is the functional side implementation requirements.

Technology stack



Desired end state - (Futuristic Perspective)

- ► The developed tool can be leveraged to integrate with large scale automation as a service cloud platform.

1. Solution should ideally suit for deployment in Azure cloud platform:
    - The tools can leverage open source/ custom libraries/ Application Program Interface (API) from cloud platforms.
    - Existing Software As A Service (SaaS)/ hosted solutions (Google Cloud Platform, Amazon Web Service etc.) except Azure should not be leveraged.
2. Preferable technology stack is C#, .net and Python for Artificial Intelligence (AI)/ Machine Learning (ML). You are also free to use libraries that are not licensed under Affero General Public License (AGPL).

It would be good if certain aspects, such as scalability, resilience, etc. can be considered as much as possible. And the solution should be modular enough so that, at some point in time, this particular cloud based solutioning is easily possible.

Q&A

1. **What is the basic input and output?**

   Ans. The basic input is the audio file. And the output will be displayed on the web page, and the specific output is just the transcript. There are two parts to the output:

1. Transcript: It is just the time stamp, the speaker ID, and what was spoken.
2. The generated minutes of the meeting.
3. Tagging: There is also one more optional step for the tagging – find and replace. If the system identifies the speaker 1, 2, and 3, you can just select them and tag them by their names. You just replace wherever those names appear, either in the meeting minutes or in the transcripts. So that is the output.

It should be a web application, where the user can interact and upload the audio. The core functionality should be exposed as an API that the web application can consume. The results also would get displayed within the web application.

2. Will they be able to get Azure service access, since it comes at a cost?

Ans. It is optional, participants will have to bear the cost themselves. We will not be providing any account for that.

3. Is it necessary to provide any downloadable format for the transcription?

Ans. No, for now, it is just enough to display it on the web page, but ideally, that response should come at the API. For example, a JSON that will be displayed on the web page so that you know we can leverage the API later on.

4. On what basis will the project be evaluated?

Ans. We will test the application with pre-recorded audio file that we will build in-house. So, we will basically check on all the functional aspects, like, how is the accuracy of the text, the speaker detection, how well it is able to extract the meeting, summary, and action items. And, brownie points might be given for additional detection. For e.g. detecting date/time for deadlines, meeting locations etc.

Overall, we will be taking a weighted sum of how well it performs on all these functionalities, and also on how flexible the solution is amenable to scalable cloud hosting.

5. **What will be the case when there are diverse languages in the meeting? What language the transcript degenerated to? And for generating a summary do we need to use a language model or any API?**

   Ans. For now, we just need to support the English language. Both the input as well as the transcripts will be in English. We are also expecting just English as the training Language model or API. The participants have the flexibility to choose a language model/ API or even combination. The only restriction is that any model you choose either should be built from scratch or open source. It should be able to be hosted on Azure platform, so that essentially rules out anything coming from other competing clouds, like Google, or AWS. But other than that, we expect open source, or custom model, or any Azure related API.

6. **Is there any specification in terms of audio format?**

   Ans. Any commonly used audio format such as MP3 or WAV is fine. We don't have any particular preference for an audio format, as long as the functional checkbox is ticked. For the evaluation, we would also see, how usable UI is on the usability for, and, how well the UI is designed, how usability tests, and so on, from a design standpoint.

7. **Will we get training data of the voice of each speaker?**

   Ans. They are expected to record their own sample audio that they can use for our model training, and so on. But towards the end for evaluation, we will test the solution with the audio sample that we have created internally. So, this sample will not be handed over initially, because, people might hard code around that, just to prevent that. Participants do have the freedom to record their own audio samples. However, they should ensure that the functional requirements are met. Initially, no other sample data would be provided.

8. **Do we need to just identify different speakers, or do we need to identify the speaker's names as well?**

   Ans. There is no need to identify the specific speaker name. You can just say speaker 1, speaker 2, speaker 3 or add any other placeholder. The actual identification of the person will be done using manual tagging. For example, the UI might provide a way for a human end user to tag, speaker 1 is so and so person, so that it gets reflected across the meeting minutes and the transcripts.

9. **For every upload, there will be a manual tag, or an algorithm should be stored, if tagged once.**

   Ans. For every upload, we expect manual tagging, but it is optional. If you don't tag, it will just be called speaker 1, speaker 2, etc. You are not expected to save tags across meeting sessions since there is no guarantee that the same speakers will come back.

10. **What is the maximum number of speakers in a given session?**

    Ans. The solution should be able to scale based on underlying platform. For the demo, the constraints of the demo device should be fine. However, if your laptop can handle only up to 5 speakers, that is okay. But if the same solution gets scaled to a bigger hosting platform like a Cloud platform, ideally, it should be able to scale, and more speakers can be added.

Disclaimer: This publication contains information in summary form and is therefore intended for general guidance only. It is not intended to be a substitute for detailed research or the exercise of professional judgment. Member firms of the global EY organization cannot accept responsibility for loss to any person relying on this article.