May 2023

# HIMEG: HIERARCHICAL MEETING NOTE GENERATION USING TEXT SEGMENTATION AND ATTENTION CORRELATION

Qixu Gong

Benjamin L Chang

Qihong Shao

# HIMEG: HIERARCHICAL MEETING NOTE GENERATION USING TEXT SEGMENTATION AND ATTENTION CORRELATION

AUTHORS:
Qixu Gong
Benjamin L Chang
Qihong Shao

## ABSTRACT

Techniques are presented herein that support a "HiMEG" system, a framework that helps create meeting notes of multiple granularities for meeting invitees so that they can refresh their memory or catch up on any meeting. Such a system comprises a Segmentation Engine that may divide a meeting transcript into separate sections representing the different topics that were covered during a meeting. Such a system also comprises an Attention Correlation Analyzing Model that may be used to capture the attention correlation between different meeting notes that were generated from the discovered topics, which is useful in a Meeting Note Summarization Model that may assess which meeting notes are most similar. Under such a system, one effective summary may be formed based on the most similar meeting notes and the process may be repeated until there is one overall summary of a meeting. In the end, a user may read the high-level summary of a meeting and then dive further into the specific contents of the general meeting note based on their interests and needs. While the above-described framework was originally developed for generating meeting notes, it may also be applied to any text input such as speeches, action scripts, and training scripts.

## DETAILED DESCRIPTION

Having a hybrid workplace within companies makes video conferences an integral part of work for employees across different industries. Oftentimes, meeting attendees rely on meeting recordings and meeting minutes to remember topics and next steps that were discussed during a meeting.

However, not everyone can attend a scheduled meeting due to being busy with work, having a conflicting meeting, or being out of the office. If an employee is too busy, or forgets, to watch the recording of a missed meeting, then they will miss potentially important information from the meeting which can be detrimental to any tasks or projects that they are working on.

The above-described problem is even worse if no one memorializes meeting minutes to share with all of the invitees. To catch up with the contents of such a meeting, an employee may need to watch the whole recording of the session even if only a small part of the meeting is relevant to their work, thus greatly reducing their work efficiency.

Figure 1, below, presents elements of an exemplary meeting script.

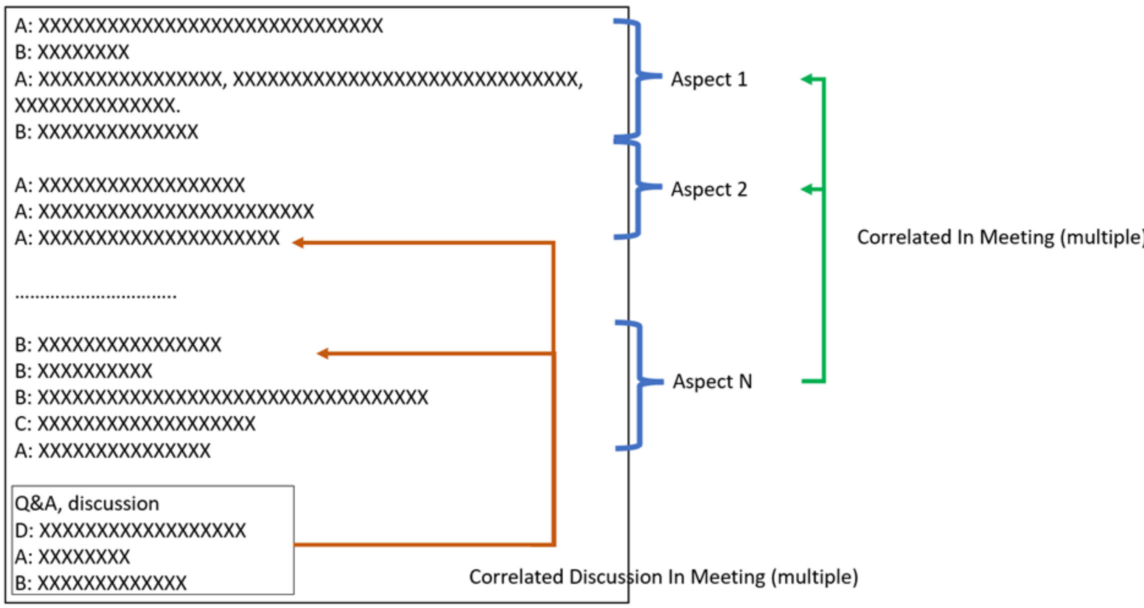


*Figure 1: Exemplary Meeting Script*

As shown in Figure 1, above, the generation of meeting notes from a meeting script is nontrivial. This difficulty may be due to, for example, many different topics and aspects being covered in one meeting; a participant's discussion of a topic or aspect being related to other topics and aspects; a meeting comprising not only its content but also its general sections or agenda such as introduction, discussion, and question and answer (Q&A) period; and the above-described sections also being associated with multiple meeting topics.

Existing meeting summarization engines leverage speech recognition techniques to translate voice into a transcript, which can range from a paragraph to a whole file. However, it can be quite challenging to search an entire transcript and locate the specific parts that an employee is interested in or that are relevant to their work.

To address the types of challenges that were described above, techniques are presented herein that support a solution that can automatically generate meeting notes on multiple levels, from one high-level summary to a summary of a specific part of a meeting. With such a solution, which may be referred to herein as a HiMEG system, employees can not only obtain an overall summary of a meeting but also specific topics of the meeting that are of interest to them.

Aspects of the presented techniques encompass a hierarchical meeting notes generation framework which can accept a meeting transcript and output meeting notes of various granularities (such as meeting- and topic-level). The framework may also be applied to text input such as speeches, action scripts, and training scripts. Further aspects of the presented techniques conduct segmentation on a meeting transcript by using machine learning to analyze dialogue sections called "turns" and automatically categorize the same by topics. Additionally, further aspects of the presented techniques conduct attention correlation by capturing semantic information from more granular meeting notes and obtaining the attention weight between them. Still further aspects of the presented techniques encompass a customized meeting note summarization model that may initially convert dialogue sections into meeting notes and then iteratively combine similar meeting notes based on attention weights.

Current natural language processing (NLP) technologies provide capabilities for summarizing text with artificial intelligence (AI) power. However, most of those capabilities are focused on the summarization of a single document. Although some models have been proposed for the summarization of multiple documents, those models only support a set of documents where the content is related across the documents and the outcome is to only generate a single summary.

Moreover, previous work has been done to create a single summary of an entire meeting from a meeting transcript. For example, one published paper employs multi-sentence compression and word importance to identify the important topics that were discussed within a meeting and create a single summary based on the same. In contrast, the techniques presented herein create multiple levels of summaries based on a user's needs.

In addition to creating one high-level summary of a meeting, through the presented techniques summaries of different topics within a meeting are also available to a user.

For segmenting meetings, previous work has been done to manually annotate segment meeting notes and divide a meeting transcript into topics that were discussed. Users would then obtain information about the meeting based on asking questions. However, a need exists for a system that automatically segments a meeting note and then develops useful information on the topics that were discussed through a set of summaries based on the granularity that a user desires. The techniques presented herein address that need by supporting the creation of a meeting note, having multiple granularities, and using machine learning to conduct segmentation and attention correlation.

Figure 2, below, presents elements of a high-level overall architecture of a HiMEG system according to the techniques presented herein and reflective of the above discussion.

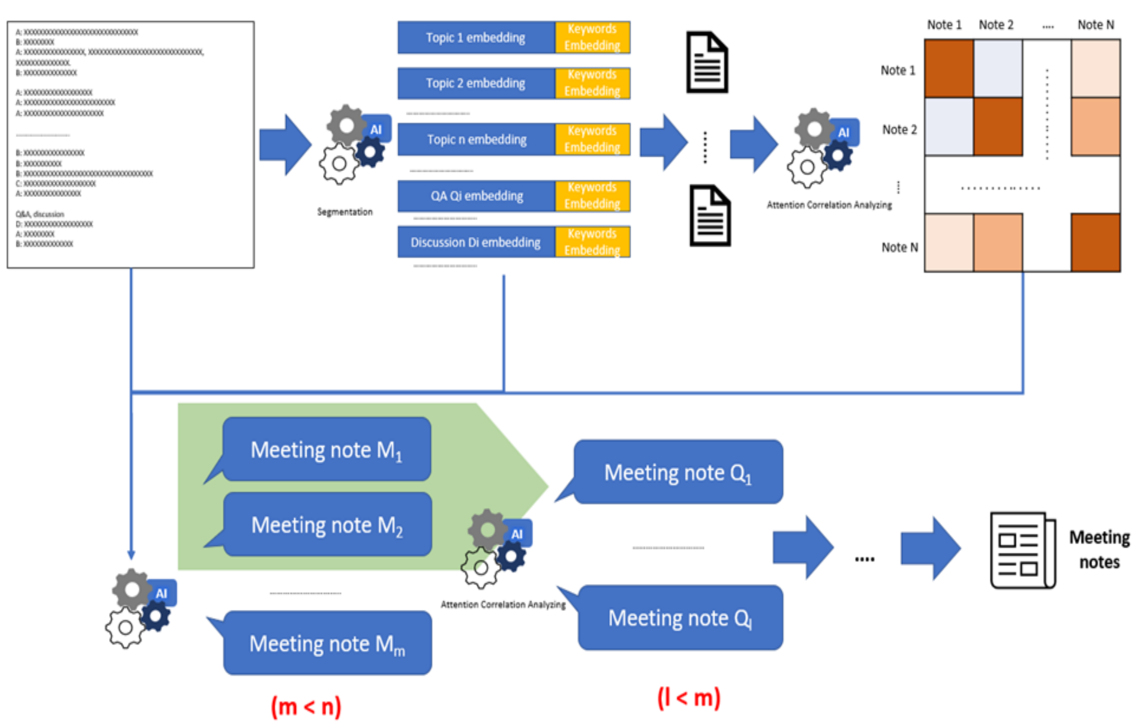


*Figure 2: High-Level Overview of HiMEG System Architecture*

As depicted in Figure 2, above, a meeting transcript may be compiled as the input to the system architecture. Such a meeting transcript may contain the speaker's name and translated text based on their spoken words.

Next, the meeting transcript may be fed into a machine learning model that conducts segmentation on the text to divide the meeting transcript into separate sections representing the different topics that were covered during the meeting. Unlike the common method of using a fixed number of topics and using a Q&A model that requires extensive training, the techniques presented herein dynamically analyze a meeting transcript and automatically segment and construct new topics based on predefined topics. From the segmentation, dialogue sections may be assigned to the extracted topics and then used during the later construction of meeting notes.

After dividing the meeting transcript into topics, NLP techniques may be used to determine the correlated content of each topic. Instead of simply conducting clustering-based methods on text-embedded information, the techniques presented herein analyze the attention correlation between multiple topics. Such an approach is useful to see which meeting notes can easily be combined to produce higher-level meeting notes.

The final portion of the system is a dedicated meeting note summarization model, which may generate meeting notes of different granularities in a hierarchical manner where each layer uses the output of all of the previous steps. Initially, topics and their related dialogue sections may be fed into the model and converted into meeting notes using NLP techniques. After conducting attention correlation between all of the generated meeting notes, the model may conduct an iteration and try to combine the meeting notes that are most correlated and form one higher-level meeting note. For example, if there are N meeting notes that were generated from a meeting transcript, then the model may input and analyze the N meeting notes and determine which ones are the most correlated. Based on that determination, some of the meeting notes may be combined to form M new meeting notes, where N is greater than M. Multiple iterations of the above-described process may be conducted until one overall meeting summary is formed that provides a high-level picture of what was discussed during the meeting.

Figure 3, below, presents elements of an exemplary HiMEG system output.

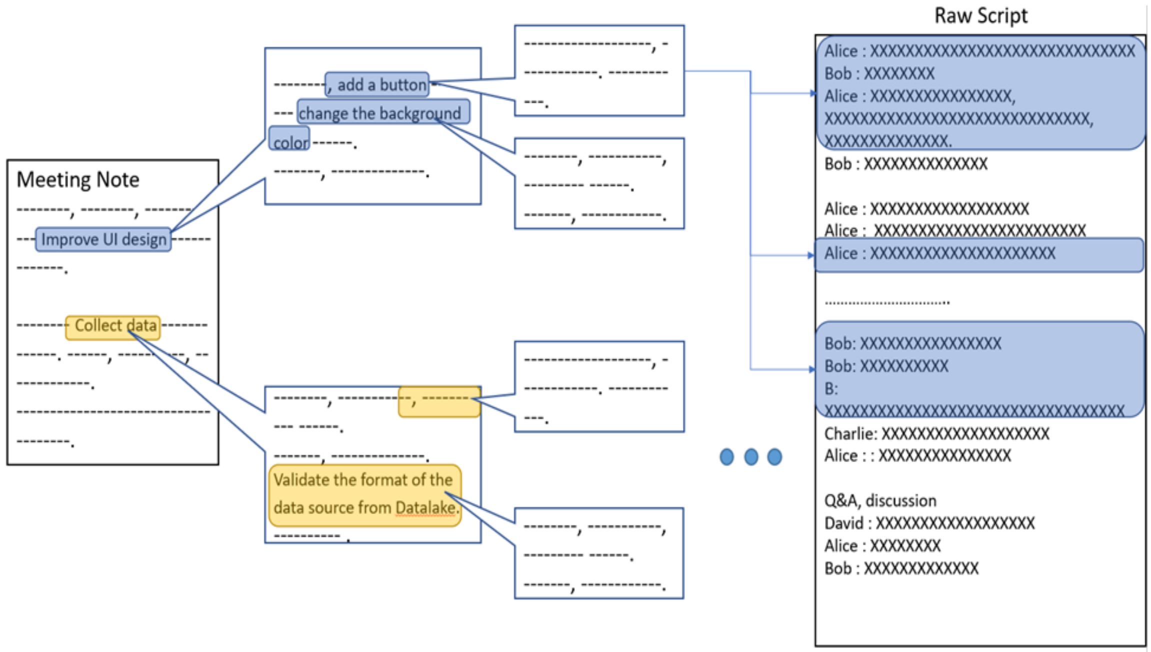


*Figure 3: Exemplary HiMEG System Output*

As depicted in Figure 3, above, a general meeting note may be generated that provides a summary of the entire meeting. A user can pursue further deep dives into the specific contents of the general meeting note based on their interests and needs. This feature enables a user to view meeting minutes based on different topics, and at different granularities, according to their needs. At the finest granularity, a user can view the original meeting transcript section that corresponds to the meeting notes that they were looking at. Overall, the end goal is to help a user refresh or catch up on meetings by letting them tailor the meeting notes to their needs.

The general meeting note that was described and illustrated above may be shown to a user to provide an overview of an entire meeting. As noted above, a user may then view more detailed meeting notes according to their needs and interests. In the example that was shown in Figure 3, above, if a user wishes to know more about user interface (UI) improvement-related content that was mentioned during the meeting, the user may then ‘click’ to jump to a more detailed meeting note about that content, which was used to summarize the previous content in the higher-level meeting note. Through iterations of looking at more detailed meeting notes, the user will be guided to the corresponding turns of a topic in the raw meeting script at the lowest level of the system.

As described above, the techniques presented herein encompass the segmentation of a meeting transcript into different topics. That activity, which will be described and illustrated in the next portion of the instant narrative, is supported by a Segmentation Engine.

Currently, the main approaches to conducting script dialogue segmentation are based on NLP Q&A models. In general, a topic or a question and the associated raw text may be input to an NLP Q&A model and then split positions are returned for each segment. However, this approach requires manually labeling a large amount of text for training and it is very costly to train and label the text. In particular, for script segmentation, a user needs to specify a fixed number of segments which greatly limits the flexibility of processing texts in the model downstream.

To address the above-described issues, the techniques presented herein support a Segmentation Engine, various of the details of which are depicted in Figure 4, below.

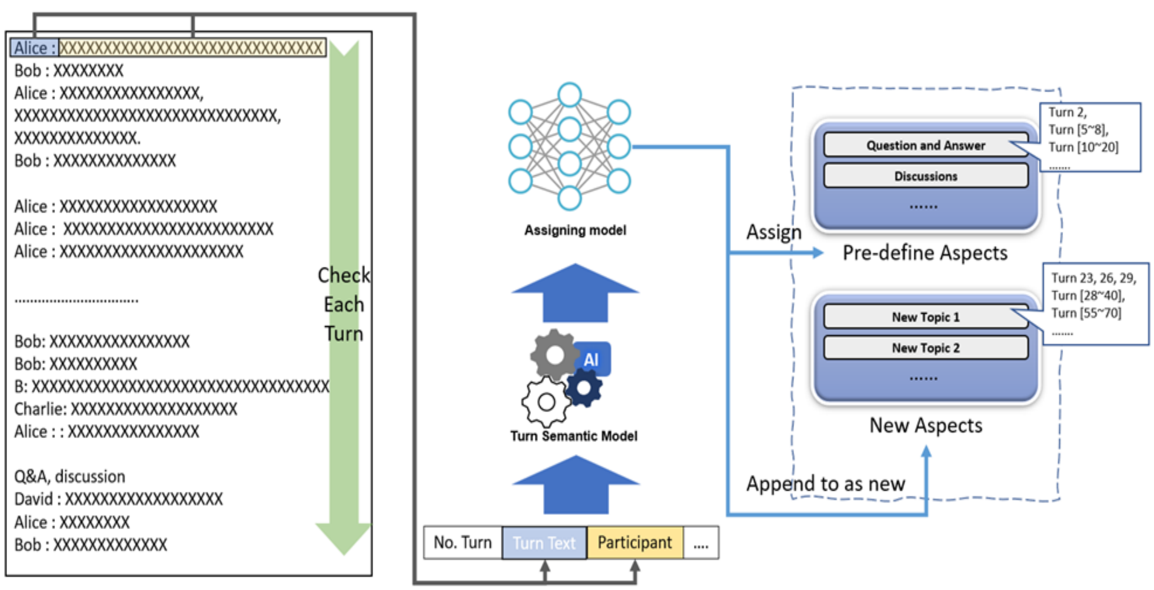


*Figure 4: Segmentation Engine Overview*

There are two types of topics that may be generated while conducting segmentation on a meeting transcript in a Segmentation Engine. One type of topic is predefined and is based on analyzing historical meeting transcript data. The other type of topic is generated while the engine scans a given meeting transcript and finds a new topic.

The term "turn" describes a dialogue section that was spoken by a meeting participant within a meeting transcript. At the beginning of the Segmentation Engine, the meeting transcript may be scanned, and the turns identified. After that, the turn number, turn text, and the speaker may be extracted and inputted into a Turn Semantic Model which embeds the information regarding each turn into high-dimensional data using a pretrained NLP model such as a Bidirectional Encoder Representations from Transformers (BERT) model. The embedding of each turn is performed to support an easy comparison between preexisting topics based on an understanding of the semantic information for each turn.

When the Turn Semantic Model compares each turn with the predefined and existing topics, one of two outcomes will occur. First, if a turn belongs to any predefined or existing topic based on the historical meeting transcript data, the corresponding turn number may be recorded, and the aggregated embedding of the topic may also be updated. Second, if there are no matches to any of the preexisting topics then a new topic may be generated for the turn and the turn number may be noted.

After processing an entire meeting transcript, all of the topics and the matching turns from a transcript may be used for further processing downstream in a HiMEG system.

As described above, the techniques presented herein conduct attention correlation by capturing semantic information from more granular meeting notes and obtaining the attention weight between them. That activity, which will be described and illustrated in the next portion of the instant narrative, is supported by an Attention Correlation Analyzing Model, elements of which are depicted in Figure 5, below.

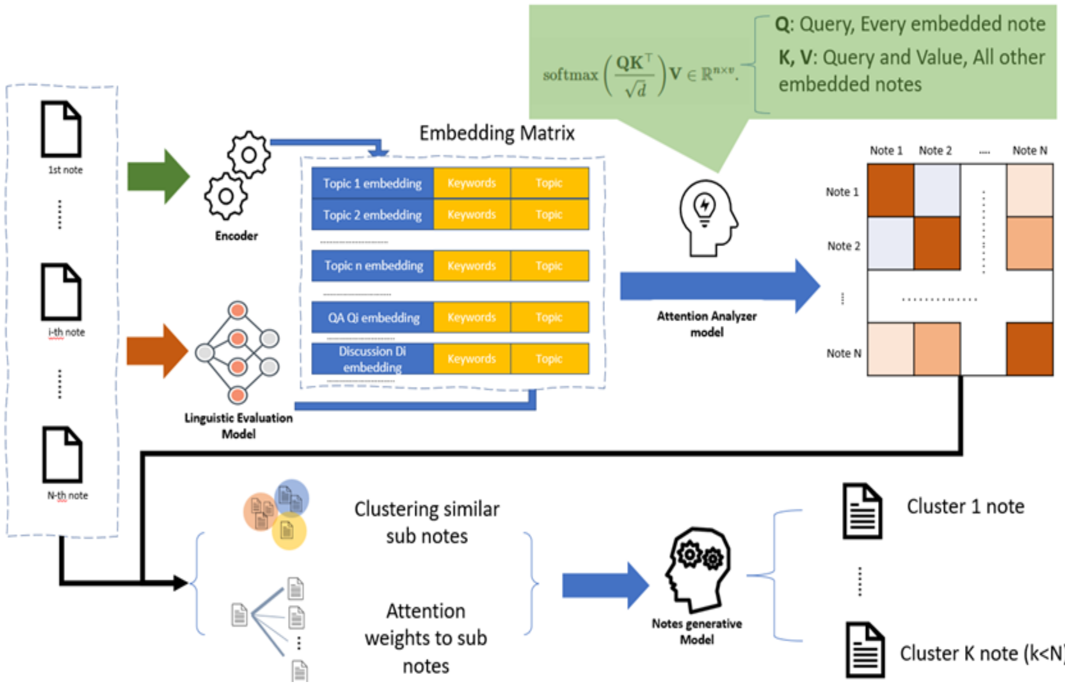


*Figure 5: Attention Correlation Analyzing Model Overview*

An Attention Correlation Analyzing Model, as shown in Figure 5, above, may accept a detailed meeting note, analyze the correlation between topics within the meeting note, and then generate higher-level meeting notes.

First, each detailed meeting note may be encoded by a pre-trained natural processing language model such as a BERT model. After that, a Linguistic Evaluation Model may be applied to better understand the semantics of the meeting note. Such a Linguistic Evaluation Model may employ a multi-task learning approach that takes the embedded text from a meeting note and outputs semantic information about each meeting note including, for example, keywords and topics. Since multiple outputs are coming from the same input, a multi-task learning model may be used instead of training individual models to gain higher accuracy on the outputs and save time and resources.

Next, an Attention Analyzer Model may be used to analyze the correlation between different topics, built based on an attention mechanism called scaled dot-product attention. An embedding of a topic may be treated as a query and all of the other topics' embeddings may be the key and value. After developing the scaled dot-product attention between topics, highly correlated topics may be assigned higher attention weights.

Following the above-described process, highly correlated topics will be clustered together based on the attention weights from the Attention Analyzer Model. The clusters may be input to a Notes Generative Model which then generates meeting notes of higher granularity based on combining similar topics. In the end, the attention weights help produce better higher-level meeting notes by incorporating information from previous meeting notes based on the magnitude of the attention weight. The model pays more attention to the input meeting notes that have higher attention weights and generates a higher-level meeting note with more content associated with the higher attention weights.

In the end, a smaller number of higher-granularity meeting notes will be generated from the Attention Analyzer Model given a list of meeting notes of smaller granularity.

The techniques presented herein leverage a hierarchical architecture, elements of which are depicted in Figure 6, below.

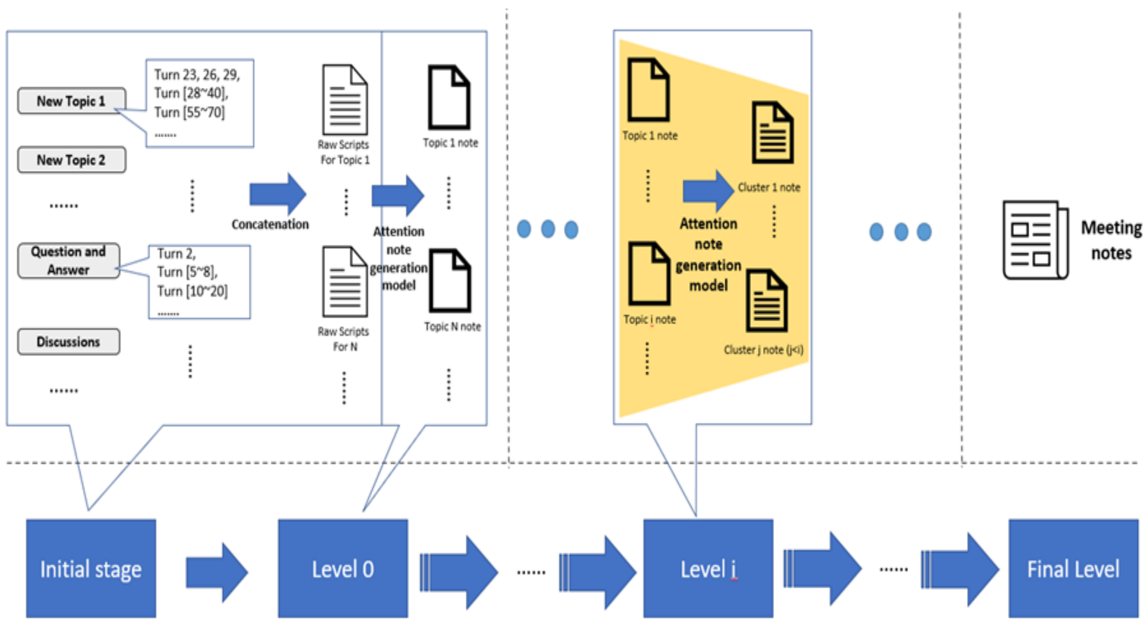


*Figure 6: Meeting Notes Summarization Model Overview*

Based on the Segmentation Engine and the Attention Correlation Analyzing Model that were described above, a Meeting Notes Summarization Model, as shown in Figure 6, above, may be introduced where each turn from a Segmentation Engine may be used to form meeting notes of higher granularity until there is just one general meeting summary for a meeting transcript. In the initial stage of the system, for each topic, the turns that were segmented may be concatenated to form raw text input. The raw text input may then be fed into the attention note generation model that generates a meeting note for each topic at Level 0.

For each subsequent level, which is depicted as Level i in Figure 6, above, the input is a set of meeting notes that were generated from the previous level that were passed to the Attention Analyzing Model, which outputs smaller amounts of higher granularity and more concise meeting notes than the previous level. Such a process may continue until one meeting note is generated that summarizes an entire meeting.

The techniques presented herein may be further explicated with reference to two use cases.

Under a first use case, Dakota (an employee of a company) is currently on paid time off for a whole week in Las Vegas and is thus out of the office regarding meetings. As a result, Dakota will miss meetings throughout the whole week, and she will need to catch up on all of her missed work and emails when she is back in the office. She will need to spend hours rewatching all of the missed meetings when time is already very scarce with work that needs to be caught up on. However, with a HiMEG system (according to the techniques presented herein) Dakota can look at the meeting notes coming from all of the meetings and quickly see a summary of each meeting along with the parts of the meeting that she is interested in. This will save her a significant amount of time which she can then spend on the work that she needs to catch up on.

Under a second use case, Taylor (another employee of a company) has back-to-back meetings scheduled during an entire day and, consequently, needs to absorb a great deal of information throughout the day. However, it is hard for Taylor to recall all of the information that was discussed during each meeting and whenever he takes notes during a meeting, he finds that it is not helpful. With a HiMEG system, Taylor can use the meeting notes to recall information within each meeting. He can read through a high-level summary to recall the purpose of a meeting and then dive deeper into the meeting notes to remember the next steps and his deliverables for a next meeting. As a result, Taylor may easily obtain all of the needed information by looking at the meeting notes compared to rewatching each recording to recall the talking points from a meeting.

In summary, techniques have been presented herein that support a HiMEG system, a framework that helps create meeting notes of multiple granularities for meeting invitees so that they can refresh their memory or catch up on any meeting. Such a system comprises a Segmentation Engine that may divide a meeting transcript into separate sections representing the different topics that were covered during a meeting. Such a system also comprises an Attention Correlation Analyzing Model that may be used to capture the attention correlation between different meeting notes that were generated from the discovered topics, which is useful in a Meeting Note Summarization Model that may assess which meeting notes are most similar. Under such a system, one effective summary may be formed based on the most similar meeting notes and the process may be repeated until there is one overall summary of a meeting. In the end, a user may read the high-level summary of a meeting and then dive further into the specific contents of the general meeting note based on their interests and needs. While the above-described framework was originally developed for generating meeting notes, it may also be applied to any text input such as speeches, action scripts, and training scripts.