Team Symantlytical @ AutoMin 2021: Generating Readable Minutes with GPT-2 and BERT-based Automatic Minuting Approach

Amitesh Garg[1]*, Muskaan Singh*[2]

[1] Thapar Institute of Engineering and Technology, [2] IDIAP Research Institute, Martigny, Switzerland

agarg11_be17@thapar.edu,msingh@idiap.ch

Abstract

This paper describes our participation system run to Automatic Minuting @ Interspeech 2021[1]. The task was motivated towards generating automatic minutes. We make a initial step towards, namely *Main Task A*, *Task B* and *Task C*. The main task A, was to automatically create minutes from multiparty meeting transcripts, while task B to identify whether the minute belongs to the transcript and task C. GPT-2[1]. The shared task, consisting of three subtasks, required to produce, contrast and scrutinize the meeting minutes. The process of automating minuting is considered to be one of the most challenging tasks in natural language processing and sequence-to-sequence transformation. It involves testing the semantic meaningfulness, readability and reasonable adequacy of the Minutes produced in the system. In the proposed work, we have developed a system using pre-trained language models in order to generate dialogue summaries or minutes. The designed methodology considers coverage, adequacy and readability to produce the best utilizable summary of a meeting transcript with any length. Our evaluation results in subtask A achieve a score of 11% R-L which by far is the most challenging than subtask as it required systems to generate the rational minutes of the given meeting transcripts.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Introduction

When most of our virtual interactions, the need for automatic support for the smooth running of online events such as project meetings became more intense. Summarizing meeting contents is one of them. Meeting minutes keep a record of what was discussed at a meeting. It is usually a written document with little or no structure(perhaps a hierarchical bulleted list) to inform the participants and non-participants of what happened during the meeting. 'Automatic minuting' tools would be a useful addition to better comprehend the meeting contents quickly. People adopt different styles when 'taking notes' or 'recording minutes' of the meeting. The minutes also depend on the category of the meeting, the intended audience, and the goal or objective of the meeting. Text or speech summarization methods known from the past would rank close to this task. However, Automatic Minuting is challenging due to the absence of agreed-upon guidelines, various minuting practices, and a lack of extensive background research. We present our submission for AutoMin [2], the first shared task on automatic minuting of meeting proceedings. Our objective is to drive community efforts towards understanding the challenges of the task and develop tools for this critical use case, especially in the current world, which has to go online far more than expected. With this shared task, we would invite the speech and natural language processing community to investigate the challenges of automatic minuting with real meeting data in two different settings: technical project meetings (both in English and Czech) and parliamentary proceedings (English).

[1]https://elitr.github.io/automatic-minuting/index.html

2. TASK A: Generating Automatic Minutes

The main task (Task A) automatically creates minutes from multiparty meeting transcripts. The generated minute would be evaluated both via automatic and manual metrics. The dataset provided for Task A, has 85 training transcripts which contain one or more summaries, 10 dev transcripts and 28 test transcripts. Every summary has corresponding annotated meeting minutes(summary) which contain the AGENDA of the meeting, as well as summaries based on the different participants and groups of the meeting. In the transcript, the speaker is indicated using curved brackets/parentheses "()", and is located at the beginning of their dialogue. Mentions of any participant or special entities are indicated using squared brackets/parentheses "[]".

There were many models deployed for the current tasks, and all of the models required different types of preprocessing. But mainly, symbols like bullets were removed from the text, and unwanted words were extracted using the list of stopwords in the NLTK library. We address this challenge and summarize a given transcript.

For this task, we chose to utilize the versatility of GPT-2[3] to suit our methodology. GPT-2(Generative Pretrained Transformer) helps us generate paraphrased human-like summaries in terms of readability. A language model is a probabilistic model that predicts the next token in a sequence given the tokens that precede it. It learns the probability of the occurrence of a sentence or sequence of tokens based on the examples of text it has seen during training. The following conditional probability can represent it:

$$P(w_1^T) = P(w_t/w_1^{t-1}) \tag{1}$$

where w_t is the t^{th} token, and writing sub-sequence be $w_i^j = (w_i, w_i + 1, ..., w_j - 1, w_j)$

GPT-2 is a variant of the transformer model which only has the decoder part of the Transformer network. It uses multi-headed masked self-attention, which allows it to look at only the first I tokens at time step t and enables them to work like traditional uni-directional language models. However, instead of processing tokens sequentially like RNNs, these models process tokens in parallel, i.e., by predicting tokens for all time steps at once. It performs the next word prediction feature of a keyboard app, but one that is much larger and more sophisticated than what your phone has. The GPT-2 was trained on a massive 40GB dataset called WebText that the OpenAI researchers crawled from the internet as part of the research effort. The smallest variant of the trained GPT-2 takes up 500MBs of storage to store all its parameters. The largest GPT-2 variant

Datasets	#Meetings	avg words per Transcript	avg words per summary	avg turns per transcript	avg # speakers
AMI	137	6,970	179	335	4
ICSI	61	9,795	638	456	6.2
Automin 120	7,066	373	727	5.9	

Table 1: *Statistics of the datasets being used in our experiments*

is 13 times the size so that it could take up more than 6.5 GBs of storage space. We have used the "GPT-2-Medium" for our task, which has around 345 million parameters. The model has a dimensionality of 1024 and has 24 layers. We have used the GPT-2 model provided by the "TransformerSummarizer," part of the "summarizer" library. This model has been pretrained on the WebText dataset[3]. WebText contains the text subset of 45 million links. A combination of the Dragnet (Peters Lecocq, 2013) and Newspaper1 content extractors was used to extract the text from HTML responses. This dataset contains about 8 million documents for 40 GB of text. A part of this dataset can be accessed here https://huggingface.co/datasets/openwebtext

2.1. Experiments and Results

GPT-2 medium is pretrained model for generating the summaries of the given transcripts. This version of the GPT-2 transformers model has around 345 million parameters. The models has 1024 dimensions and has 24 layers. Some other hyperparameters presented in Table 2 include a number of layers as 24, summary type as 'cls-index,' number of attention heads for each attention layer in the transformer encoder as 16, and the dropout probability for all fully connected layers in the embeddings, encoder, and pooler as 0.1. A lot of models were compared and evaluated before this model was chosen. The models included both extractive and abstractive models. The summaries generated by these models were compared with the summaries given in the training set using 'Rouge-1' (the number of matching unigrams) and 'Rouge-L' (quantification of similairty based on the longest matching subsequence). The GPT-2 model was found to be performing best, and according to the hardware available, this GPT-2 medium was chosen. We evaluate our results with ROUGE, or Recall-Oriented Understudy for Gisting Evaluation is used to calculate the quality of the summary produced by the particular machine model. It does so by measuring the number of overlapping text units (i.e., n-grams) between model generated and ground truth summaries. We used the metrics of rouge, R-1(1-gram rouge) and R-L(Longest common subsequence rogue) to get the evaluations. This similarity method was used for Task A as summaries were compared. We present our results in Table 5 automatic evaluation, the final Rouge-1 score achieved was 0.2167 and Rouge-l Score achieved was 0.1111, and by human evaluation in which an average of 2 annotator scores were generated, the average scores(Likert scale of 1-5, 5 being the highest) for all the meetings were: Fluency: 2.4643, Adequacy: 2.9821 and Grammatical Correctness: 2.6428.

Table 2: *Common Hyperparameter Details for Models Used*

Hyperparameter	TaskA	Task B	Task C
Attention Probabilities Dropout Ratio	0.1	0.1	0.1
Vocabulary Size	50257	30522	50265
LayerNorm Epsilon	1e-5	1e-12	1e-5
Model Type	GPT2	BERT	ROBERTA
Python	3.7		
GPU Type	Nvidia Tesla K80		
GPU RAM	12 GB		
M/c Ram	12 GB		
CPU	Intel(R) Xeon(R)(Virtual) 2.0Ghz		

3. Task B and Task C: Semantic Similarity

In Task B, Given a pair of meeting transcripts and a minute, the task is to identify whether the minute belongs to the transcript. During our data preparation from meetings on similar topics, we found that this task could be challenging given the similarity in various named entities. In Task C, Given a pair of minutes, the task is to identify whether the two minutes belong to the same or different meetings. This sub-task is important as we want to uncover how minutes created by two different persons for the same meeting may differ in content and coverage. In task B, the challenge is given a transcript and minutes of some meeting; whether the minutes belong to the particular transcript has to be decided. The task helps in deciding the similarity in both the documents and will help determine whether the minutes and transcripts are of the same meeting or not. Our method utilizes a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model[4], in which the key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modeling. This contrasts with previous efforts that looked at a text sequence from left to right or combined left-to-right and right-to-left training.

A big part of this is Bert's ability to embed the meaning of words into densely packed vectors. We call them dense vectors because every value within the vector has a value and has a reason for being that value — this is in contrast to sparse vectors, such as one-hot encoded vectors where the majority of values are 0. BERT is great at creating these dense vectors, and each encoder layer (there are several) outputs a set of dense vectors.

The model we have used is named "bert-base-nli-mean-tokens" [5], which is provided by hugging face. In this model, the BERT base has been used, creating the dense vectors containing 768 values. These 768 values include our numerical representation of a single token — which we can use as contextual word embeddings. Because there is one of these vectors representing each token (output by each encoder of BERT), we are looking at a tensor of size 768 by the number of tokens.

We can take these tensors — and transform them to create semantic representations of the input sequence. We can then take our similarity metrics and calculate the respective similarity between different sequences. Let's say that the number of tokens is 512, then the major steps performed will be:

- Creating the Vector.
- To convert the final obtained tensor into our vector — we use a mean pooling operation.
- Each of those 512 tokens has a respective 768 values. This pooling operation will take the mean of all token embeddings and compress them into a single 768 vector space — creating a 'sentence vector'.
- Then apply cosine similarity between these sentence vectors of the transcripts and the minutes to get a similarity value.

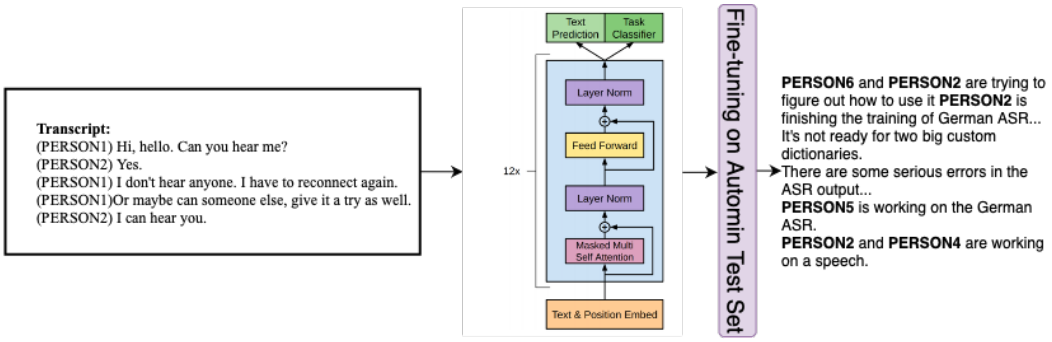

Figure 1: *Proposed Architecture to generate autoamtic minutes*

As for data preprocessing, we took all the transcripts and minutes, removed all the stopwords present in the "NLTK" library, and removed all the unwanted symbols from the text.

Table 3: *Dataset provided for task B*

Dataset	True Label	False Label	Total
Train+Dev	115	728	843

This model was originally trained on SNLI[6] and MultiNLI datasets. Stanford Natural Language Inference(SNLI) database is a collection of sentence pairs labeled for entailment, contradiction, and semantic independence. At 570,152 sentence pairs, SNLI is two orders of magnitude larger than all other resources of its type. And in contrast to many such resources, all of its sentences and labels were written by humans in a grounded, naturalistic context. We collected four additional judgments for each label in a separate validation phase for 56,941 of the examples. The dataset can be found here: https://nlp.stanford.edu/projects/snli/ The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated textual entailment information. The corpus is modeled on the SNLI corpus. Still, it differs in that it covers a range of spoken and written text genres and supports a distinctive cross-genre generalization evaluation. The dataset can be found here:[2] In task C, the challenge is given two different minutes of some meeting; it has to be decided whether the minutes belong to the particular transcript. The task helps in deciding the similarity if both the documents and will help determine whether the minutes are of the same meeting.

Sentence-BERT (SBERT) is a modification of the BERT network using siamese and triplet networks that can derive semantically meaningful sentence embeddings. This enables BERT to be used for certain new tasks, which were not applicable for BERT until now. RoBERTa builds on BERT's language masking strategy, wherein the system learns to predict intentionally hidden sections of text within otherwise unannotated language examples. RoBERTa, implemented in PyTorch, modifies key hyperparameters in BERT, including removing BERT's next-sentence pretraining objective and training with much larger mini-batches and learning rates. This allows RoBERTa to improve the masked language modeling objective compared with BERT and leads to better downstream task performance. SBERT adds a pooling operation to the output of BERT / RoBERTa to derive a fixed-sized sentence embedding.

Specifically, for the experiment, we used 'paraphrase-distilroberta-basev1'[5], which is a 'DistilBERT-base-uncased' model fine-tuned on a large dataset of paraphrase sentences. This RoBERTa-based sentence representation model has been trained to produce meaningful sentence embeddings for similarity assessment and retrieval tasks. It uses a vector length of 768 for the sentence embeddings. After getting the sentence embeddings, the cosine similarity metric was applied to get the similarity value. Finally, cosine similarity was used on the sentence embedded vectors to get a similarity value between them.

As for data preprocessing, we took all the minutes, removed all the stopwords present in the "NLTK" library, and removed all the unwanted symbols from the text.

Table 4: *Dataset provided for task C*

Dataset	True Label	False Label	Total
Train+Dev	116	818	934

This model was trained on millions of paraphrases which included ALLNLI, which is a combination of SNLI[6] and MultiNLI datasets. Stanford Natural Language Inference(SNLI) database is a collection of sentence pairs labeled for entailment, contradiction, and semantic independence. At 570,152 sentence pairs, SNLI is two orders of magnitude larger than all other resources of its type. And, in contrast to many such resources, all of its sentences and labels were written by humans in a grounded, naturalistic context. In a separate validation phase, we collected four additional judgments for each label for 56,941 of the examples. The dataset can be found here: https://nlp.stanford.edu/projects/snli/ The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated textual entailment information. The corpus is modeled on the SNLI corpus. Still, it differs in that it covers a range of genres of spoken and written text and supports a distinctive cross-genre generalization evaluation. The dataset can be found here: [3]

[2]https://cims.nyu.edu/ sbowman/multinli/

[3]https://cims.nyu.edu/ sbowman/multinli/

Teams	Adequacy	Grammatical Correctness	Fluency	ROUGE-1	ROUGE-2	ROUGE-L
Average of all teams	2.81	3.25	2.92	0.203	0.0458	0.114
Ours	**2.98**	**2.64**	**2.46**	**0.2167**	**0.044**	**0.111**

Table 5: *Results for subtask A on test data. The adequacy, grammatical correctness, and fluency are evaluated manually by two annotators and assessed on a Likert Scale of 1 to 5. These scores are based on official results provided by the organisers and are averaged across all test set samples.*

Table 6: *Task- B (Test-Set (k=7))*

System	Accuracy	Precision	Recall	F1
Ours	0.948	0.94	0.95	0.94
Majority	0.944	-	-	-

		Accuracy
Task B	Our submission	42.6%
	Average	0.944
Task C	Our submission	
	Average	0.936

Table 7: *Accuracy of approaches used by our team for both Task B & C on validation data compared to average*

3.1. Experiments and Results

To perform this classification, the similarity values are produced on the embedding produced by a pre-trained model, and then a threshold is used to achieve the binary classification. The pre-trained model used is "bert-base-nli-mean-tokens" provided by hugging face. In this model, BERT-base has been used, which creates the dense vectors containing 768 values. These 768 values contain our numerical representation of a single token — which we can use as contextual word embedding. Some other hyperparameters for this model include the non-linear activation function (function or string) in the encoder and pooler as 'gelu', the number of attention heads for each attention layer in the transformer encoder as 12 and the standard deviation of the truncated-normal-initializer for initializing all weight matrices as 0.02. After this model produces the embeddings, cosine similarity (measured similarity between two feature vectors by capturing the document's orientation and not the magnitude, unlike the Euclidean distance) is measured, and the final similarity values are produced. Many threshold values are checked to minimize the mismatching of actual binary classifications and the generated binary classifications. The final threshold value is chosen by 0.65. The final scores yield an accuracy of 42.6%. A pretrained model was used to obtain the sentence embedding. Then a similarity metric was used to get the similarity values, and finally, a threshold value was obtained to get the final binary classification. The pretrained model used is "paraphrase-distilroberta-basev1', which is a 'DistilBERT-base-uncased'model fine-tuned on a large dataset of paraphrase sentences. This RoBERTa-based sentence representation model has been trained to produce meaningful sentence embedding for similarity assessment and retrieval tasks. It uses a vector length of 768 for the sentence embeddings. Some other hyperparameters for this model include the non-linear activation function (function or string) in the encoder and pooler as 'gelu', the number of attention heads for each attention layer in the transformer encoder as 12 and the standard deviation of the truncated-normal-initializer for initializing all weight matrices as 0.02. After this model produces the embeddings, cosine similarity is measured, and the final similarity values are produced. Many threshold values are checked to minimize the mismatching of actual binary classifications and the generated binary classifications. The final threshold value is chosen by 0.65. The final scores yield an accuracy of 79.8% in Table: 7.

—

4. Conclusions

In this work, we have described the system we used for the AutoMin@Interspeech 2021 on automatic minuting and analysis and comparison of meeting minutes. There were three tasks according to which different methods were applied for the particular tasks. We used all the pre-trained models for the evaluation as they have very high accuracy and are well-known models for their particular use cases. We have also discussed various models that can be used for these specific tasks. However, automatic minuting is still a challenge for various reasons, like diversity in minuting practices and minuting done by models limited to the trained data. However, the pre-trained models have been extensively trained, and they can be fine-tuned and used for efficient minuting approaches.

5. References

[1] V. RISNE and A. SIITOVA, "Text summarization using transfer learnin: Extractive and abstractive summarization using bert and gpt-2 on news and podcast data," 2019.

[2] T. Ghosal, O. Bojar, M. Singh, and A. Nedoluzhko, "Overview of the first shared task on automatic minuting (automin) at interspeech 2021," in *Proceedings of the First Shared Task on Automatic Minuting at Interspeech 2021*, 2021, pp. 1–25. [Online]. Available: http://dx.doi.org/10.21437/AutoMin.2021-1

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[6] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.

6. Generated Samples

Following is an example of minutes generated by our model sampled from the data for Task A:

DATE : 2021-07-21
ATTENDEES : PERSON4, PERSON5, PERSON8, PERSON10, PERSON13

SUMMARY-

- The deadline for the project is next Monday, June 15th.
  Someone from the project needs to be registered there.
  PERSON8 will try to register today.
- PERSON13 is going with PERSON4 to LOCATION5.
  They have a meeting before lunch on Monday.
  They have one more paper, she wants to submit it to Archive and PROJECT8 so that someone can read it.
- PERSON10 is on holiday for next two days.
  They have written one and half paragraph of the book yesterday, and will work on the book from now on.
- PERSON4 will write half of the chapters.
- PERSON8 will organize the chapters.
  They added some information from papers.
  They will write a preface to the book.
  He needs to generate, to get the similar metrics from the PROJECT3 and the rest.
  https://www.overleaf.com/project/60f83a18c950b85c4d9e99b2
- PERSON5 is going to write his survey.
  They will work with PERSON8.
- ALL are working on the papers.
  The deadline for feedback is at the end of June.
  The reviewers for PROJECT5 need to be at least a professor, but don't have to be from the university.
  The grant will be 5000 for it.
  The deadline for PROJECT7 should be in November.
  The conference will be virtualised and take place in 2021.
- PERSON8, PERSON13, PERSON5 and PERSON10 discussed the details of the conference.
  The abstract submission is on Monday, June 15th.
  PERSON5 and PERSON8 are going to write a survey for the project. They want to introduce new people to it.
- ALL discussed about the amount of money they are getting from the university.
  The money for this year cannot be used for bonuses.
  PERSON7 bought the computer that he is now using for some grant.
- PERSON8 got a mail from PR person saying that they can come to the official event.

No need for this sample And, on the following page there is a true positive instance predicted by our model, for TASK-C :

Minute:A)

- (PERSON9) Oh, uh, uh, we're waiting for someone, maybe, uh, that is going to join us, but I'm not sure.
- (PERSON4) Uh, not sure, I'll check this, because, uh, I didn't see it at first. ¡laugh¿
- (PERSON9) Oh, you actually call it [LOCATION2] time?
- (PERSON2) Like simply said, it's 1 hour from now and 4 days and 1 hour from now. And also, uh, [PERSON6]asked me, uh, to tell you [PERSON4] that you s– you should think about the possibility that –Basically, the moment the travel restrictions are, uh, going d– go down. And, uh, the way they employed you, uh, is really, like, they, they had to close both of their eyes. Uh, and, uh, it is important for you to come in person to, uh, enter a proper work contract. And, uh, it would be better for us and better for the project if you decided to take it as an opportunity and not as a, as a negative must. Or I'm looking at it, at we are still waiting for How, how is it with [ORGANIZATION4]? So, I again, the same situation, I don't know whether they started it or not.
- (PERSON9) We can definitely raise that on the Thursday call, maybe it ne–The thing that the communication is not completely perfect and, uh. It, it's sort of like a –
- (PERSON2) Like it i– like it is fine, but, uh, yeah, uhm. I think that, uh, the, the bullet point is quite explanation Like, what's the term. Like, uh, is there any, was there any issue, something that needs to be discussed? Because they might be sensitive to domain shift and, and such, right? So, can try do both and check the differences and, and stuff. Like, right, uh, I guess with paraphrasing or if we use it for example for paraphrasing, we might use the metric to, uh, estimate the sim– sim– simil– like semantic similarity, but still have the constraints to enforce some, some like surc– surface form. (PERSON2) Uh, maybe just like my stupid question, why do we need this eva– new evaluation for MT? (PERSON9) No, so, I, I can –
- (PERSON2) Why do we need like another like non–standard metric? And the thing is that we want to kinda guarantee a semantic similarity, right, which BLEU doesn't capture, because it's just using the surface forms during evaluation.
- (PERSON9) Yeah, this is – this definitely sounds interesting, because again, I think that the papers that do focus on the constraint decoding, they, they really don't evaluate it from this perspective, right? Ye–, uh, well but then I think like this, this might be a good, good approach to the, to the task. But yeah, we can, we can wrap it up, and I guess see you, see you next week, right?

Minute:B)

Organizational stuff

- Monthly call will be on Thursday, 5 PM LOCATION1 time
  - At least PERSON14 and PERSON10 should take part
  - PERSON14 will care about including PERSON6 into the mailing list
- PERSON6's coming to LOCATION1
  - It is very desirable that PERSON6 comes to LOCATION1 in person
  - Visa issues due to Covid situations

PROJECT2

- PERSON10 is trying to contact ORGANIZATION5 colleagues, the communication is not completely perfect
- PERSON4 is preparing the leaflets, LOCATION1 is waiting

Progress on PROJECT6

- PERSON10 is trying the back-translation
  - It's low priority, is running on server, but may be stopped if needed.
  - No interesting results to discuss yet. Should be discussed with PERSON15 first, what to do next
  - PERSON10 may try the translations on CPUs

PROJECT4

- No special updates for now
- a related paper on BLEU that might be useful for evaluation
- Discussing metrics, using semantic metrics, different kinds of metrics
- Why do we need special metrics for MT
