Team ABC @ AutoMin 2021: Generating Readable Minutes with a BART-based Automatic Minuting Approach

Kartik Shinde[1]*, Nidhir Bhavsar*[2]*, Aakash Bhatnagar*[2]*, Tirthankar Ghosal*[3]

[1]Indian Institute of Technology Patna, India
[2]Navrachana University, Vadodara, India
[3]Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Czech Republic

kartik_1901ce16@iitp.ac.in, 18103488@nuv.ac.in, 1824526@nuv.ac.in, ghosal@ufal.mff.cuni.cz

Abstract

This paper documents the approach of our Team ABC for the First Shared Task on Automatic Minuting (AutoMin) at Interspeech 2021. This challenge's primary task (Task A) was to generate meeting minutes from multi-party meeting proceedings automatically. For this purpose, we develop an *automatic minuting* pipeline where we leverage a denoising autoencoder for pretraining sequence-to-sequence models and fine-tune it on a large-scale abstractive dialogue summarization dataset to summarize meeting transcripts. Specifically, we use a BART model and train it on the SAMSum dialogue summarization dataset. Our pipeline first splits the given transcript into blocks of smaller conversations, eliminates redundancies with a specially-crafted rule-based algorithm, summarizes the conversation blocks, retrieves the block-wise summaries, cleans, structures, and finally integrates the summaries to produce the meeting minutes. Our proposed system performs the best in several evaluation metrics (automatic, human) in the AutoMin shared task. We use certain text similarity metrics for the subsidiary tasks to determine whether a given transcript-minute pair corresponds to the same meeting (Task B) and if a given pair of meeting minutes belong to the same meeting (Task C). However, our simple machine-learning-based approach did not perform well in addressing the objective of the subsidiary tasks in the challenge. We publicly release our system codes here[1].

Index Terms: automatic minuting, meeting summarization, topic segmentation, multi-party dialogues

1. Introduction

Ever since the COVID-19 pandemic hit our lives, most workplace interactions have gone virtual. Hence, the demand for automatic support for the smooth running of online events and meetings has grown more than ever. Frequent remote conferences and day-to-day workplace video calls demand an efficient system to document these interactions concisely to reduce the cognitive overload on the participants resulting from simultaneous note-taking during the meeting or retrieving/recalling information from past meetings. Hence meeting summarization is a practical and timely problem for the speech and natural language processing community. However, the task is not straightforward as it may seem. Many factors can affect the automatic minuting quality, including the employed Automated Speech Recorder (ASR) quality, participant's background, real-time update summarization on long dialogue discourse, retaining topical relevance, privacy, ethical issues, etc. However, these challenges do not diminish the need for such a workplace innovation. Although 'Meeting Summarization' and 'Automatic Minuting' appear to be the same problem, certain subtle differences exist. While summarization is motivated towards generating a concise and coherent summary of the text, minuting is more inclined towards adequately capturing the contents of the meeting (*where coverage is probably more significant than coherence and conciseness*). The AutoMin shared task at Interspeech 2021 is one event dedicated to invoking and steering relevant research on this topic. In this paper, we present our system run at the AutoMin shared task [1]. Our system came out to be one of the **best-performing systems** on several human and automatic evaluation metrics (especially on ROUGE-1, ROUGE-2, ROUGE-L in automatic metrics, and Fluency, Grammatical Correctness in human evaluation metrics). Table 1 shows the task-specific goals for this shared task.

Main task A: **(Generation)**	Transcript → Minute
Subtask B: **(Verification)**	Transcript + Minute → True/False (True corresponds to a pair of minute and transcript from the same meeting, else false)
Subtask C: **(Comparison)**	Minute + Minute → True/False (True corresponds to a pair of minutes belonging to the same meeting, else false)

Table 1: *Task-specific challenges in the AutoMin shared task*

The main task in the AutoMin challenge is *Automatic Minuting*. There is no universal framework for creating minutes, and the activity varies across different types of meetings, subjects, and agendas/objectives. Hence, one person may create a minute that is very different from another person for the same meeting (the reason for instantiation of Task B and Task C in

[1]https://github.com/cruxieu17/automin-2021-submission

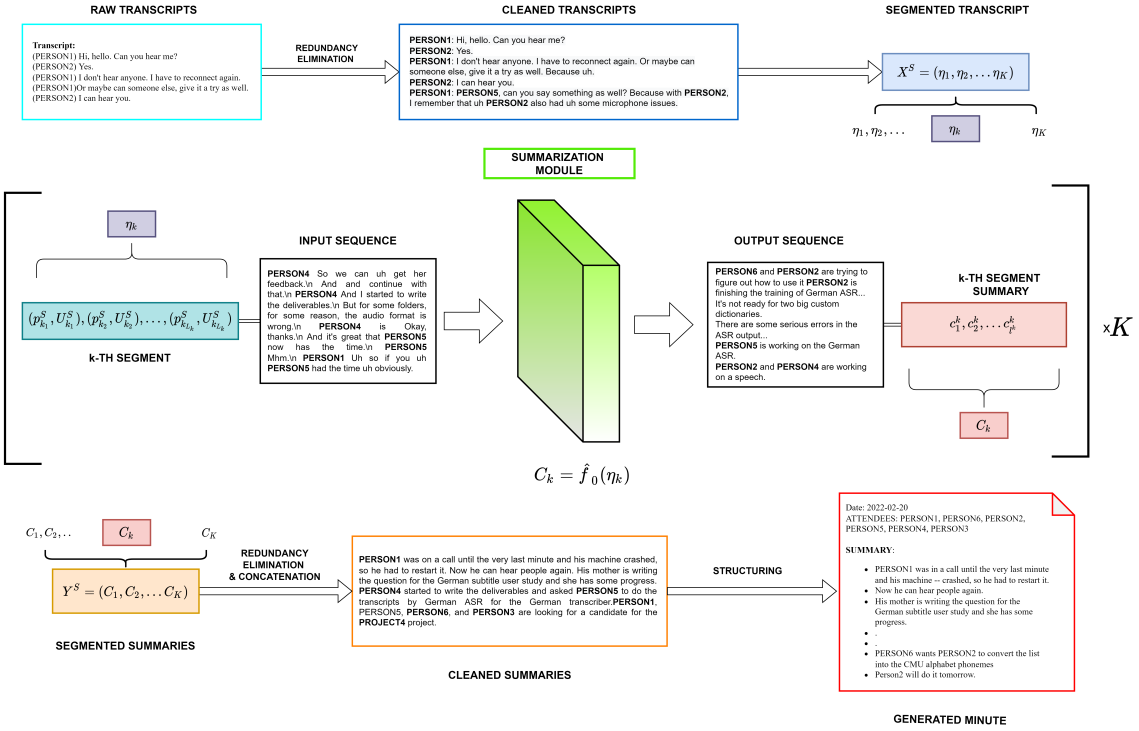

Figure 1: *Architectural representation of our Automatic Minuting pipeline (Task-A).*

AutoMin). This variation can be in terms of - format, length, use of novel words, ignorance towards trivial details, writing style, etc. Hence, to evaluate the different facets of summaries, apart from the popular automatic ROUGE scores [2], the AutoMin challenge laid more stress on human judgments where a candidate minute is evaluated based on its adequacy, readability, semantic meaningfulness, clarity, coverage, and grammatical correctness. Although generating bullet-point minutes from multi-party meetings is a relatively novel one, there have been efforts toward multi-party meeting summarization in the literature. We review and take inspiration from a few of those. Early studies like Chen and Metze, (2012) [3] used intra-speaker topic modeling to improve summarizing multi-party meetings. In 2019, a few approaches came from Zhao et al. (2019) [4]; Liu and Chen, (2019) [5]; Liu et al., (2019) [6] which revealed the efficacy of hierarchical methods to learn the inherent structure of conversations. Li et al., (2019) [7] demonstrated a multi-modal hierarchical attention mechanism across the topic, utterance, and word levels. However, it depends on manual annotation of topical segments and visual focus in meetings that are not commonly available. We were inspired by these works to use topic segmentation in our approach. However, given the numerous challenges in developing an efficient and cost-effective end-to-end training framework, we adopt a pipelining approach for the main task. As shown in figure 1, our proposed pipeline begins with extensive rule-based text processing, followed by abstractive summarization and redundancy elimination using the TextRank algorithm [8], and finally, chronologically structuring of minutes to generate a concise and adequate minute from a meeting transcript.

Optional subtasks B and C involved analyzing and comparing meeting minutes and transcripts and identifying similarities and coherence among a pair of meeting minutes OR a transcript and a minute. For tasks B & C, we first vectorize the texts. We extract various features by combining different similarity measures popularly used to estimate the relatedness between pairs of text documents. Specifically, we use similarity scores like ROUGE-1, 2, and ROUGE-L, Jaccard Similarity, Cosine Similarity, BERTscore, etc. These scores can supposedly capture both semantic as well as syntactic meaning. Further, we train classification models like Random Forest [9] & Support Vector Machine [10] on multiple combinations of the extracted features and then choose the best performing classifier as the final model. However, the F1 score and accuracy figures were not ideal due to significant data imbalance.

2. Task A: Automatic Minuting

2.1. Overview

As is evident, Task A is the primary objective of this shared task. We select the BART model pre-trained on the xsum dataset, and further fine-tuned on SAMSum dataset [11] as our primary summarization module. We also experiment with other well-known Transformer-based encoder-decoder models like T5[12],

Pegasus[13], and RoBERTa2RoBERTa [14]. We compare the performance of the pipeline with different hyperparameter configurations and underlying summarization modules and finally use the optimal setup as our final submission.

2.2. Dataset Description

The AutoMin corpus for Task-A has 85, 10, and 28 instances for train, dev, and test sets, respectively. Each instance comprises (i) a meeting transcript; (ii) one or more than one annotated meeting minutes corresponding to the given transcript. Table 2 and 3 show several relevant statistics of the datasets that we have used in experimentation. Table 4 shows the statistics of the AutoMin data that were made available to us in the shared task.

Datasets	# Dialogues	# Turns	# Speakers	# Avg. Turn Len.
SAMSum	16.4K	11.2	2.4	9.1
AutoMin	124	254.4	5.8	9.7

Table 2: *Statistics of the dialogue and meeting datasets used in our experiments.*

Datasets	# Len. of Dialogue	# Summary Len.	%-Compression
SAMSum	124	23.4	81.12%
AutoMin	8,890.8	387	95.65%

Table 3: *Some further statistics of the dialogue and meeting datasets used in our experiments*

Even though the annotated minutes do not conform to a particular format, we observe that almost every minute consists of several information points - the date of the meeting, attendees, agenda, bulleted minutes, and annotator of the corresponding minute. Minutes for a single meeting transcript annotated by several annotators might not be coherent or consistent. However, as per the minuting guideline stated in AutoMin [1], the generated minutes should be concise and concrete, avoid overuse of person names, and should majorly emphasize topical coverage, action points, and decisions. Also, bullet points are preferred over coherent textual summaries for generated minutes.

	Train	Dev	Test-I	Test-II
Task A	85	10	18	10
Task B	565	280	972	-
Task C	555	378	1431	-

Table 4: *Number of instances in AutoMin Shared Task data across the 'EN' task and dataset splits*

2.3. Methodology

As shown in Figure 1, our pipeline is divided into three major modules. We begin with analyzing various meetings and perform preprocessing of the input text, redundancy elimination, segmentation, and abstractive summarization. The next part of our pipeline is a summarization module. Finally, we filter the output using an unsupervised redundancy elimination method to obtain the minutes.

2.3.1. Redundancy Elimination

Existing summarization models are not trained to eliminate redundancies from a long dialogue discourse. Besides this, they are also capped to specific input lengths for precise and good-quality text generation. Hence, they struggle to process a longer sequence of multi-speaker utterances and the dispersed information that comes with them. BART's training approach helps it learn to handle noisy inputs more efficiently with respect to other models. We employ text processing for utterance-cleaning and redundancy elimination using hand-crafted rules.

As shown in the topmost part of Figure 1, a raw transcript with Speaker-Utterance pairs, $X^0 = \{({p^0}_1, {U^0}_1), ({p^0}_2, {U^0}_2), ..., ({p^0}_L, {U^0}_L)\}$, where ${p^0}_j \subset P$, $1 \leq j \leq L$, is a participant and ${U^0}_j = ({w^j}_1, {w^j}_2..., {w^j}_{l_j})$ is the tokenized utterance from p_j. For i-th utterance, ${U^0}_i = ({w^i}_1, {w^i}_2..., {w^i}_{l_i})$ in the transcript, we generate a cleaned sequence, ${U^c}_i = ({W^i}_1, {W^i}_2..., {W^i}_{L_i})$, by eliminating repetitions, pauses and masks like {vocalsounds}, {disfmarkers}, {unintelligible} and similar disruptions. These utterances are then filtered using a custom stopwords set, $S = \{s_1, s_2, ...s_n\}$, that we define with the help of various meeting transcripts from publicly available meeting summarization corpuses like AMI [15], ICSI [16], and the dataset provided in the shared task. Following this, we obtain the compressed utterance, $U^x = U^c \cap S'$ and corresponding context ratio, R as shown.

$$R = L(|U^x|)/L(|U^c|) \tag{1}$$

here, L(|A|) is the length of set 'A'. Finally, a processed transcript is obtained by appending the utterances whose context ratios qualify a certain threshold α.

$$X' = \sum_{i=1}^{L} [(p_i, {U^c}_i) | R_i \geq \alpha] \tag{2}$$

2.3.2. Linear Segmentation

As seen in 1, we adopt a brute-force approach here by slicing the transcripts into blocks of segments with a uniform token length. We do this to address the limitation of the current deep neural summarization models to input a full-length transcript. Our motivation here is to maintain the adequacy of the generated minutes while covering every necessary detail across the transcripts; hence we did not want to get restricted by the input limitations of the summarization models. However, while carrying out experiments, we observed that one must calculate a threshold to limit the number of tokens in a segmented block of conversation, which would help minimize the effect of contextual interdependence across distant blocks in a conversation. Hence, we adopt varying token lengths of 512, 768, and 1024, respectively, in our pipeline.

2.3.3. Summarization

We choose the pretrained BART model, from [17], as the primary summarization module in our pipeline. Also, we

experiment with summarization models using T5, Pegasus, RoBERTa2RoBERTa, etc. However, the BART-based pipeline performed better than the rest. We fine-tune all our summarization models on well-known dialogue summarization datasets (especially SAMSum [11]) before integrating them into our pipeline. For finetuning the underlying summarization module, we use the following configurations: 'max_input_length' = 512, 'min_target_length' = 128. As mentioned, the training data for this method is the SAMSum corpus [11]. We use a learning rate of $2e^{-5}$ and a batch size equal to 4 during the training.

BART[17] is a denoising autoencoder for pretraining sequence-to-sequence models. The model is trained by corrupting text by an arbitrary de-noising function and then teaching it to reconstruct the original text. BART's ability to use bi-directionalism when operating on sequence generation tasks makes it useful for abstractive text summarization. While BERT [18] cannot adopt a bidirectional mechanism for sequence generation, BART exploits the GPT-2 [19] architecture for predicting the following words with the help of words encountered previously in the current sequence. Hence, we primarily test the pipeline with various BART-based setups.

We pass the input sequence obtained from the preprocessing module through the summarization module. For the k-th segment, it returns a summary $C_k = \{c^k{}_1, c^k{}_2, ...c^k{}_{l^k}\}$, where $c^k{}_i$ is the i-th summary line of the k-th segment. We rejoin all the segment summaries $Y^S = (C_1, C_2, ...C_K)$ to get the raw summary text.

2.3.4. Post-processing

To eliminate the redundancies in the generated outputs, we use a similar process as in section 2.3.1. Since the models are fine-tuned to provide fluent outputs, another elimination procedure seems needless. These summaries present an adequate amount of information but lack somewhat in *Adequacy*. For instance, the model might capture a casual conversation from some part of the meeting that mentions a particular location the speaker traveled to or some other miscellaneous detail. To tackle this issue, we use TextRank [8]. On average, the model captures around 15% trivial and irrelevant information from the full-length summary. We rank the summary lines in increasing order of their importance (or contextual richness) and exclude out bottom 15% of the lines to obtain a "gold span" of the summary, which we further sort and format to obtain more concise minutes. To further compress the summaries, we add appropriate pronouns, eliminate grammatical inconsistencies wherever possible, and filter the final chain of conversation threads by excluding unnecessary words using a stopword set that we internally develop by observing the generated summaries.

2.4. Results

Table 5 shows our submission results on the auto minuting task. Our submission received an Adequacy of 4.25 out of 5, placing us on par with the second-best performing system while performing much higher (almost double) than the average score of all other teams. Doing better in the adequacy criteria signifies that our methods can adequately capture various interrelated topics within the discussions. Moreover, our results were superior to any other team's submission in terms of *Grammatical Correctness* and *Fluency*. Our system also achieved 28.2 points on ROUGE-1, which exceeded the scores of all other systems. We also outperform the other systems in terms of ROUGE-2 and ROUGE-L.

The high *Grammatical Correctness* scores signify the merit of our selection of the BART model fine-tuned on the SAMSum corpus. The generated minutes are consistent in terms of *Fluency* and *Adequacy*. Although our system performs comparatively better, the 6.55 points on ROUGE-2 show that the generated minutes are not congruent to the reference minutes. One reason is that the reference minutes from the same meeting instance fail to match each other in terms of the information content and format due to annotations from different annotators.

2.5. Analysis and Discussion

The results clearly show that our system efficiently processes the meeting transcripts and can generate fluent, concise, and adequate minutes. Given the instance-wise automatic and human evaluation scores, we analyze their correlation to understand better the differences in the two evaluation strategies and what these scores signify. One meeting transcript has been annotated with one or more than one minute by different annotators. Hence, the evaluation provides two scores - 'max' and 'avg', which denote the maximum and average of the automatic system-generated minute scores against each human-created minute. In our case, since the maximum instance-wise human evaluation (*Fluency*, *Grammaticality*, *Correctness*) scores of most of the generated instances lie in the upper 10% of the Likert Scale (i.e., 4.5-5 out of 5), the variation in the scores is significantly less. Hence, we choose the average score over the maximum score to better understand the variation of all the metrics relative to one another.

Figure 2 shows the correlation between the automatic and human evaluation scores using the 'Pearson Correlation Coefficient'. There is a lesser correlation between automatic and human evaluation metrics signifying the necessity of human judgments in the evaluation.

3. Task B & C: Pair Classification

3.1. Overview

Task-B aims to identify whether a minute and a transcript belong to the same meeting instance in a given minute-transcript pair. The gist of this task lies in understanding the differences between two document pairs (transcripts-minutes) and determining how similar their content is. Sub-task C aims to identify whether or not a given pair of meeting minutes corresponds to the same meeting transcript. As discussed earlier, a given transcript, minuted separately by multiple annotators, can result in

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	**4.25**	**4.45**	**4.27**	**0.282**	**0.0655**	**0.159**

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.*

Models	R-1*	R-WE	BLEU	BERT**	TF-IDF
BART	0.297	0.162	2.907	0.563	0.19
DistilBART	0.375	0.205	6.535	0.620	0.25
T5	0.406	0.229	6.278	0.615	0.31
Ours	**0.45**	**0.298**	**7.068**	**0.673**	**0.38**

Table 6: *Performance of different baseline models considered during experimentation. *R denotes the ROUGE score, **BERT denotes BERT-F1 score.*

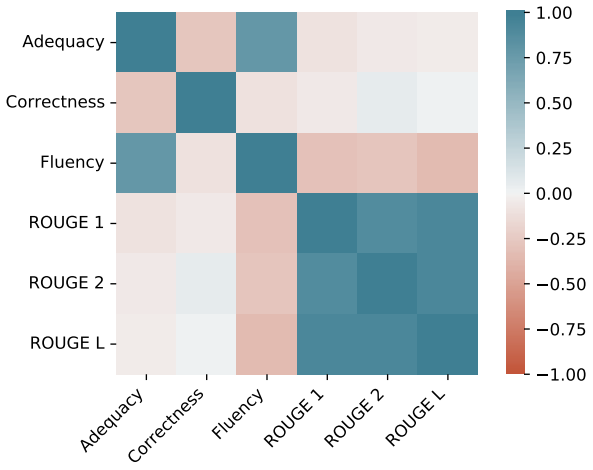

Figure 2: *Correlation between the evaluation metrics used in the shared task.*

significantly different minutes in terms of the information content, note-taking style of the different annotators with different backgrounds, and their structure.

3.2. Datatset Description

Task-B's dataset comprises transcript-minute pairs and corresponding labels: i.e., TRUE and FALSE. Here, TRUE indicates that the minute matches with the paired transcript and vice versa. As annotations done by different annotators can vary in terms of writing style, grammar, tone, vocabulary, etc., the minutes curated by each annotator are different and have a unique style/format. On the other hand, the dataset in Task-C comprises meeting minute pairs and their corresponding labels. Here, TRUE indicates that the paired minutes correspond to the same meeting transcript and vice versa. Both these data have a good imbalance between the instances belonging to the two classes (as represented by Table 7). In both cases, the TRUE is the minority class; merely 15% representation in the entire data.

Dataset	True tag	false tag	**Total**
Task B	115	731	**846**
Task C	74	660	**734**

Table 7: *Class-wise distribution of train and dev data.*

3.3. Pre-processing

We refine the data by removing all undesired characters through Spacy's POS tagging tool[2] and filtering unnecessary information from the text. The training data in tasks B & C have distinctly visible class imbalance, with the FALSE class in predominance. Due to this imbalance, our classifiers' recall and precision values were significantly low. To mitigate this, we choose to perform oversampling on the minority class from the dataset. For these two tasks, we derive scores for each extracted pair of texts based on various methods that we describe in the subsequent section and generate a R^5 dimensional vector corresponding to each instance. Moreover, we generate an equal amount of positive and negative samples to balance the data via oversampling.

3.4. Similarity Scores

We list the various similarity scores we use for Task B and Task C. These scores act as features to the input and are often quite distinct, contributing to the alignment of pairs of minutes or minutes and transcripts. For some scores, we require the input text in vectors to represent the text documents. We first experiment with pre-trained embeddings and models but eventually consider TF-IDF vectors. Hence our approach primarily relies on the word-level features rather than the semantic ones.

3.4.1. Cosine Similarity

The cosine similarity measures the similarity between two vectors via inner product. It is measured by the cosine of the angle between two vectors and determines whether two text articles/documents are pointing in roughly the same direction. We considered using the cosine similarity pairwise metric to compute the similarity between the text documents.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{3}$$

3.4.2. ROUGE-N & ROUGE-L

The **R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation (ROUGE) metric enables us to calculate the similarity between

[2]https://spacy.io/usage/linguistic-features

a pair of meeting minutes as well as meeting minutes and transcripts. Moreover, ROUGE-1 to be precise allows us to measure the overlap in uni-gram obtained by filtering stop-words, whereas ROUGE-L instead measure the overlap between the longest running sub-sequence within both text documents under consideration.

$$\text{rouge-n(X, R)} = \frac{\sum_{r_i \in \text{R}} \sum_{\text{n} \in r_i} \text{count(n, X)}}{\sum_{r_i \in \text{ref}} \text{num(n)}\,(r_i)} \quad (4)$$

Here elements r_i are sentences in the reference document ref, $count(n, X)$ is the number of times the specified n-gram n occurs in the candidate document X and $num(n)(r_i)$ is the number of n-grams n in the specified reference sentence r_i.

$$\text{rouge-l (X, R)} = \frac{\left(1+\beta^2\right) R_{\text{lcs}}(\text{ X, R })P_{\text{lcs}}(\text{ X, R })}{R_{\text{lcs}}\,(\text{X, R }) + \beta^2 P_{\text{lcs}}(\text{ X, R })} \quad (5)$$

Usually, the F-score is taken for the ROUGE-L metric between a candidate document X and a single reference document R. Here the parameter β controls the relative importance of the precision and recall. Because the ROUGE score favors recall, β is typically set to a higher value.

3.4.3. *Sequence Matcher*

The main objective to use *SequenceMatcher*[3] is to find the longest contiguous matching sub-sequence LCS with no "irrelevant" elements. The characters that we don't want the algorithm to match are irrelevant, like blank lines in ordinary text files, etc. This metric does not yield minimal edit sequences but does tend to yield matches that logically seem appropriate.

3.4.4. *Jaccard Similarity*

With Jaccard similarity, we measure the similarity of two meeting minutes in terms of their content, i.e., how many common words are compared to the total number of words. Here J is the Jaccard distance calculated via the distinct word present in set A and B.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

3.4.5. *BERT score*

BERT-Score[20] takes advantage of BERT's pre-trained contextual embeddings and uses cosine similarity to match words in candidate and reference documents. It has been demonstrated that it correlates with human judgment on sentence-level and document-level evaluations [4].

3.5. Model

For experimentation and evaluation purposes, we use the available classifier models in the SciKit-Learn library [21]. We found that with our set of features, the Support Vector Machine (SVM) and the Random Forest Classifier outperform the other classifiers.

[3] https://docs.python.org/3/library/difflib.html
[4] https://github.com/Tiiiger/bert_score

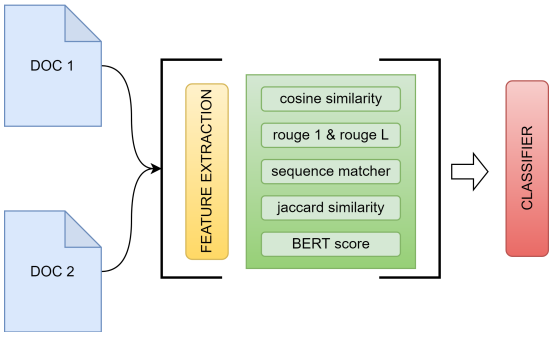

Figure 3: *The overall architecture of our proposed system for Subtask B and C*

	Classifier	Accuracy	Precision	Recall	F1
Task B	Random-Forest	**0.91**	**0.71**	**0.62**	**0.66**
	SVM	0.88	0.65	0.40	0.49
Task C	Random-Forest	**0.85**	**0.42**	**0.61**	**0.5**
	SVM	0.77	0.26	0.53	0.35

Table 8: *Our results for Task B & C on the validation data.*

3.6. Results

Table 8 describes the two main approaches employed by our team for the alignment of pairs of minutes (Task C) and minutes-transcripts (Task B). Our results with the Support Vector Machine (SVM) over the derived features were poor, but the Random-Forest classifier, which fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve predictive accuracy, helped in performance. However, these results are not as good as the best-performing systems. Our current system lacks sufficient training data (See Table 7), and even after oversampling, the model's compatibility with newly available data in the validation set remained low.

4. Conclusion and Future Work

In this work, we describe our system run for the first AutoMin shares task on automatic minuting and analysis/comparison of meeting minutes. Our proposed system leverages large-scale transformer-based language and summarization models to generate readable minutes from multi-party meeting proceedings. We ranked one of the top-performing systems for the main task (Task A) of automatic minuting. Our proposed automatic minuting approach scores are among one of the top approaches in automatic as well as human evaluation metrics. In the future, we would like to implement an unsupervised topical segmentation strategy instead of the current brute-force one. We also plan to train a joint multitasking model by incorporating all the subtasks in our pipeline.

5. Acknowledgement

Tirthankar Ghosal has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 825460 for the project European Live Translator (ELITR)[5].

6. References

[1] 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

[2] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[3] Y.-N. Chen and F. Metze, "Integrating intra-speaker topic modeling and temporal-based inter-speaker topic modeling in random walk for improved multi-party meeting summarization," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[4] Z. Zhao, H. Pan, C. Fan, Y. Liu, L. Li, M. Yang, and D. Cai, "Abstractive meeting summarization via hierarchical adaptive segmental network learning," in *The World Wide Web Conference*, 2019, pp. 3455–3461.

[5] Z. Liu and N. Chen, "Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5460–5466.

[6] Z. Liu, A. Ng, S. Lee, A. T. Aw, and N. F. Chen, "Topic-aware pointer-generator networks for summarizing spoken conversations," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 814–821.

[7] M. Li, L. Zhang, R. J. Radke, and H. Ji, "Keep meeting summaries on topic: Abstractive multi-modal meeting summarization," in *57th Conference of the Association for Computational Linguistics*, 2019.

[8] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 404–411. [Online]. Available: https://aclanthology.org/W04-3252

[9] M.Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005. [Online]. Available: https://doi.org/10.1080/01431160412331269698

[10] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[11] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "Samsum corpus: A human-annotated dialogue dataset for abstractive summarization," *arXiv preprint arXiv:1911.12237*, 2019.

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[13] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 328–11 339.

[14] S. Rothe, S. Narayan, and A. Severyn, "Leveraging pre-trained checkpoints for sequence generation tasks," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 264–280, 2020.

[15] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The ami meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88. Citeseer, 2005, p. 100.

[16] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 1. IEEE, 2003, pp. I–I.

[17] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.

[18] 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.

[19] 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.

[20] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with BERT," *CoRR*, vol. abs/1904.09675, 2019. [Online]. Available: http://arxiv.org/abs/1904.09675

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *CoRR*, vol. abs/1201.0490, 2012. [Online]. Available: http://arxiv.org/abs/1201.0490

[5] https://elitr.eu

7. 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.

Minuted by: Team ABC
