Team UEDIN @ AutoMin 2021: Creating Minutes by Learning to Filter an Extracted Summary

Philip Williams, Barry Haddow

School of Informatics, University of Edinburgh, Scotland

philip.williams.edin@icloud.com , bhaddow@ed.ac.uk

Abstract

We describe the University of Edinburgh's submission to the First Shared Task on Automatic Minuting. We developed an English-language minuting system for Task A that combines BERT-based extractive summarization with logistic regression-based filtering and rule-based pre- and post-processing steps. In the human evaluation, our system averaged scores of 2.1 on adequacy, 3.9 on grammatical correctness, and 3.3 on fluency.

Index Terms: automatic minuting, extractive summarization, meeting summarization

1. Introduction

The University of Edinburgh participated in the main task of the First Shared Task on Automatic Minuting [1]. We developed a pipelined system that employs (more-or-less) off-the-shelf extractive summarization together with rule-based and learned components. The output of our system is a short list of bullet points (roughly 3% the length of the original transcript) together with a list of participants. The bullet points are sentences derived from the original transcript, but cleaned up to remove speech disfluencies and transcription artifacts. Figure 1 shows a sample of the resulting minutes.

While automatic minuting is a form of meeting summarization, minutes typically emphasize certain aspects of the meeting, such as decisions that have been reached or actions that are to be taken. We found that extractive summarization alone produced mixed results in terms of selecting sentences that are appropriate for use in minutes. We therefore employed a post-summarization step that filtered the summarizer output. To do this we first hand-labelled a sample of summarizer output for the training data then trained a logistic regression model to score extracted sentences according to their 'minute-worthiness.'

Ultimately, we did not make use of the minutes provided in the training data except as a guide for making system design choices. This was due in part to the wide variety of minuting styles used by the annotators, with wide variations in minute length as well as structural choices, such as grouping bullet points by topic or by speaker. This diversity of styles made the data challenging to utilize effectively for machine learning and is an aspect of minuting that sets it apart from other summarization tasks.

Submissions to this task were evaluated automatically using ROUGE [2] and manually using human judgment of adequacy, fluency, grammatical correctness. In the human evaluation, our system averaged scores of 2.1 on adequacy, 3.9 on grammatical correctness, and 3.3 on fluency. During system development we evaluated minuting quality on the dev set using ROUGE-1 and ROUGE-2 scores computed using sacreROUGE [3] and we report those scores in this paper. In practice, we found that ROUGE scores calculated against the supplied references were not sufficiently reliable to differentiate systems and we relied on manual sample checking for making most model design choices.

```
Attendees: PERSON1, PERSON2, PERSON3,
PERSON4, PERSON5

 * [PERSON1]: I plan to go there, but like,
   we need a back-up person.
 * For the [PROJECT2] event.
 * We need someone to take care of the
   recording, so the archiver person.
 * [PERSON3]: I think we need to improve
   our segmenter, the worlds are getting
   revised fine.
 * [PERSON3]: I'll first ask him to correct
   the current [PROJECT6]L for the correct
   type we have.
 * Maybe it will be better for us to attend
   the call with the [PERSON7].
 * We will separately need to ship the
   audio to the English [PROJECT5]
   separately.
 * [PERSON1]: If you have good data for the
   language pair, then yes, it is better to
   go directly.
```

Figure 1: *Sample system output (test meeting 27).*

2. The Automatic Minuting Pipeline

Figure 2 outlines the pipeline that transforms a raw transcript into minutes. This section describes the individual steps in detail.

2.1. Preprocessing

The preprocessing step of the pipeline performs three main tasks: it normalizes speaker attributions, records the list of participants, and removes speech artifacts.

2.1.1. Speaker Attribution

The raw transcripts contain a speaker attribution, such as `[PERSON5]`, at the start of each turn. Since summarization will be performed at the sentence-level, we copy speaker attributions to the start of each sentence in order that attributions will persist through the summarization and filtering steps (although we may later choose to discard some of them). The training data uses a mix of square and round brackets, which we standardize as square brackets.

2.1.2. Participant List

The list of participants is recorded at this stage since later steps may remove the contributions of some speakers.

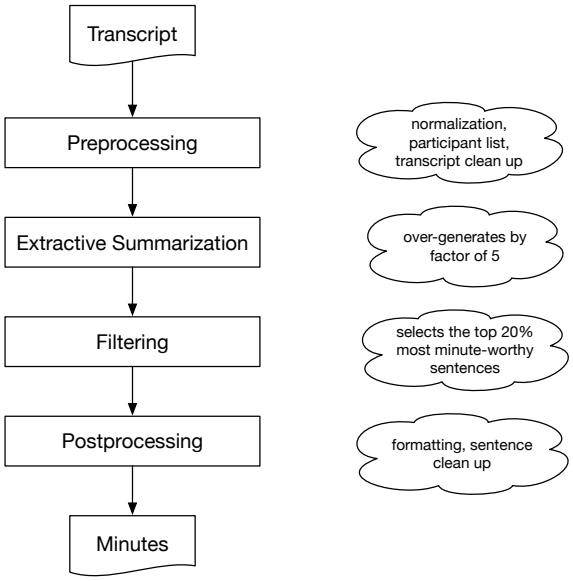

Figure 2: *The automatic minuting pipeline.*

2.1.3. Removal of Speech and Transcription Artifacts

The raw meeting transcripts faithfully reproduce speech disfluencies as well as adding annotations in the form of tags, such as `<laugh>` or `<other_language>`. We use hand-written rules to repair sentences where possible and to remove sentences that are incomplete. Specifically, we:

- Remove filler words, such as `um`, `er`, and `ehm`;
- Repair restarts such as, `we sh-, we should`;
- Remove sentences containing the `<unintelligible>` tag
- Remove incomplete words (ending with the – character);
- Remove incomplete sentences (not ending `.` or `?`); and
- Remove any remaining annotation tags.

For example, after preprocessing, the sentence,

```
[PERSON3]: will send uh, SLT findings uh,
findings PDF. <parallel_talk>
```

becomes,

```
[PERSON3]: will send SLT findings PDF.
```

The removal of incomplete and unintelligible (or partially unintelligible) sentences constitutes a significant level of filtering prior to the main summarization and filtering stages, reducing the total number of sentences by approximately 28%

2.2. Extractive Summarization

For summarization, we used `lecture-summarizer`[1] [4] with minor modifications. `lecture-summarizer` is an extractive summarizer based on the BERT [5] pre-trained language model. It is designed to summarize transcripts of university lectures. In brief, it works by using BERT to encode sentences, clustering the sentence embeddings, and then finding the nearest sentence to the centroid of each cluster. The number of clusters is configurable and can be specified either as a fixed number of sentences or as a ratio. We chose a ratio of 0.035, which is toward the lower-end of transcript / minute ratios in the training data. We chose the exact value based on personal preference, since the ROUGE metric was an unreliable guide, almost always favouring longer minutes.

[1] `https://github.com/dmmiller612/lecture-summarizer`

2.2.1. Modifications to `lecture-summarizer`

`lecture-summarizer` uses spaCy [6] to split the input into sentences. We found that the spaCy sentencizer would separate speaker attribution tags from sentences, and since our transcripts had already been split into sentences, we modified the code to use line breaks as sentence delimiters instead.

In order that we could over-generate sentences (ahead of the subsequent filtering stage), we modified the code to produce the k closest sentences to each centroid (k was set to 5 in our final system).

2.3. Filtering

In preliminary systems, we noticed that the output of the summarizer would typically contain a small number of sentences that were perfect to include in the minutes as-is, among a larger number of sentences that were unsuitable, because they were irrelevant, vague, or lacked context. While the suitability of many sentences is borderline or subject to opinion, it was clear that there were some features that could differentiate the most suitable sentences from the least. We therefore tried hand-labelling a set of sentences produced by the summarizer from the training and dev meetings and training a regression model to score candidate sentences.

In total, we labelled 1,107 sentences from the training meetings, assigning 266 (24.0%) to the positive class (minute-worthy) and 841 (76.0%) to the negative class. We labelled 451 sentences from the dev meetings, assigning 176 (41.9%) to the positive class and 244 (58.1%) to the negative class.

For model development, we created a balanced training set containing all 266 positive examples and 266 randomly sampled negative examples from the training meetings. Similarly, we created a test set containing all 176 positive examples and 176 randomly sampled negative examples from the dev meetings.

We used scikit-learn [7] to train a logistic regression model with unigram and bigram TF-IDF features. On our test set, this achieved a precision of 66.7%, recall of 65.9% and F1 score of 66.3%. Figure 3 gives a sample of low- and high-scoring sentences from our test set.

In the pipeline, we used this model to score candidate sentences produced by the summarization step, taking the top-scoring 20%

2.4. Postprocessing

Postprocessing performs three final tasks that are primarily stylistic: it removes conjunctions and exclamations from the starts of sentences, selectively drops speaker attributions, and formats the participant list and summary.

2.4.1. Conjunction and Exclamation Removal

Spoken sentences frequently begin with a conjunction, such as `so` or `because` or with an exclamation such as `yeah` or `oh`. We remove these when they occur at the start of a sentence, based on a list of words observed in the output of the training data.

```
0.03 So does it work, if you are not searching
     for any words?
0.06 I, maybe I just didn´t compile it
     properly.
0.07 So probably it´s not that serious.
0.08 but I think, you saying, you've got an
     email from a project officer?
0.10 I 'm not, you know, wanting is like,
     yes?

0.90 So I agree with what [PERSON6] suggested
     that [PERSON4] and [PERSON12] should
     focus on the selection of the input.
0.92 and we have the reviewer chosen.
0.92 This week I work on do the collection is
     business for [OTHER1] and English.
0.95 I still have to look at it and then we
     have to prepare for the posters.
0.98 We will buy some extra time for from
     them.
```

Figure 3: *A sample of low-scoring and high-scoring sentences, as scored by our logistic-regression model.*

2.4.2. Speaker Attribution Removal

In order that the minutes appear less like direct speech, we remove the speaker attributions for any sentence that does not include a reference to the first or second person (`I`, `me`, `your`, etc.).

2.4.3. Formatting

Finally, we format and output the list of participants that was saved during preprocessing and we add bullet points to the summary.

3. Experiments

Here we give results for our submitted system in contrast to some baseline and variant systems that were created during system development. We used ROUGE for evaluation, since that is the task's primary automatic metric, although, as already mentioned, we found it to have limited use during system development.

The systems are as follows:

`baseline-random` Randomly selects sentences from the transcript. No pre- or postprocessing except for speaker attribution normalization, bullet points, and participant list generation.

`baseline-lecsum` As `baseline-random` but uses `lecture-summarizer` to select sentences.

`submitted` Submitted system

`no-filter` As `submitted`, but does not include filtering step and does not over-generate during extractive summarization.

Table 1 gives average ROUGE-1 and ROUGE-2 scores on the dev set for systems tuned to produce output of approximately the same length. In the case that multiple references were available for a meeting, we computed scores against all references and took the maximum. Note that this differs from the official evaluation method, which takes the average.

While the submitted system is the highest-scoring (on ROUGE-2), the differences in score are small and we found that substantial differences in quality between system were not reflected in the scores.

System	ROUGE-1	ROUGE-2
`baseline-random`	27.8	5.0
`baseline-lecsum`	29.6	5.8
`submitted`	29.4	6.5
`no-filter`	26.0	5.4

Table 1: *ROUGE-1 and ROUGE-2 scores on the dev set.*

4. Discussion

The minutes produced by our system give a sense of what a meeting was the about and tend to include at least some of the actions and outcomes. However, the minutes are unsatisfactory in a number of important ways, largely resulting from the use of extractive summarization:

- Many sentences lack context and are unable to stand alone;
- The minutes contain direct speech where reported speech would be more natural;
- There is no means for the system to encapsulate portions of the meeting in a single sentence (e.g. '[PERSON4] and [PERSON7] discussed arrangements for the upcoming conference');
- The minutes are unstructured.

In an attempt to address the problem of sentences lacking context, we experimented with coreference resolution, using the neuralcoref[2] package to replace corefering mentions with main mentions. However, we found that harmful substitutions (where correct terms were replaced with incorrect ones) were more common than beneficial substitutions.

Before developing our current system, we briefly experimented with abstractive summarization (specifically with the Pegasus model [8] in Huggingface [9]). Abstractive summarization is appealing for this task and would potentially solve at least some of the problems listed above. It has been successfully applied to meeting summarization for the AMI [10] and ICSI [11] datasets [12]. However, it was unclear to us how to address some significant challenges posed by this dataset, most notably, the diversity in minuting styles and the length of the transcripts (even the Longformer[13] model available in Huggingface 'only' supports an input of 4,096 tokens, which is far short of the meeting transcript lengths).

In a first attempt to make the data more amenable to learning with a transformer-based model, we began chunking the transcript and manually aligning bullet points, with the goal of creating smaller training examples, but found this was difficult in practice for many of the minutes due to the extreme summarization and restructuring of material in the minutes.

We also attempted to segment the transcripts into parts that could be tackled separately during inference. We experimented with the NLTK implementation of TextTiling [14] but on inspecting results, it didn't appear to pick up meaningful boundaries.

[2] `https://github.com/huggingface/neuralcoref`

5. Conclusion

We have described the University of Edinburgh's submission to the First Shared Task on Automatic Minuting. Our minuting system was based on extractive summarization with logistic regression-based filtering and rule-based pre- and post-processing steps. While our system performed satisfactorily in terms of grammatical correctness and fluency, it performed less well in terms of adequacy, which we attribute to the use of extractive summarization.

6. Acknowledgments

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 825460 (ELITR).

7. 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] D. Deutsch and D. Roth, "SacreROUGE: An Open-Source Library for Using and Developing Summarization Evaluation Metrics," in *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 120–125. [Online]. Available: https://www.aclweb.org/anthology/2020.nlposs-1.17

[4] D. Miller, "Leveraging BERT for extractive text summarization on lectures," *CoRR*, vol. abs/1906.04165, 2019. [Online]. Available: http://arxiv.org/abs/1906.04165

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[6] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020. [Online]. Available: https://doi.org/10.5281/zenodo.1212303

[7] 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," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," 2019.

[9] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[10] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus: A pre-announcement," in *Machine Learning for Multimodal Interaction*, S. Renals and S. Bengio, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 28–39.

[11] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The icsi meeting corpus," 2003, pp. 364–367.

[12] X. Feng, X. Feng, and B. Qin, "A survey on dialogue summarization: Recent advances and new frontiers," *CoRR*, vol. abs/2107.03175, 2021. [Online]. Available: https://arxiv.org/abs/2107.03175

[13] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020.

[14] M. A. Hearst, "Texttiling: A quantitative approach to discourse segmentation," *Computational Linguistics*, vol. 23, no. 1, pp. 33–64, mar 1997.