CUED at ProbSum 2023: Hierarchical Ensemble of Summarization Models

Potsawee Manakul, Yassir Fathullah, Adian Liusie,
Vyas Raina, Vatsal Raina, Mark Gales
ALTA Institute, Engineering Department, University of Cambridge
{pm574,yf286,al826,vr313,vr311,mjfg}@cam.ac.uk

Abstract

In this paper, we consider the challenge of summarizing patients' medical progress notes in a limited data setting. For the Problem List Summarization (shared task 1A) at the BioNLP Workshop 2023, we demonstrate that Clinical-T5 fine-tuned to 765 medical clinic notes outperforms other extractive, abstractive and zero-shot baselines, yielding reasonable baseline systems for medical note summarization. Further, we introduce Hierarchical Ensemble of Summarization Models (HESM), consisting of token-level ensembles of diverse fine-tuned Clinical-T5 models, followed by Minimum Bayes Risk (MBR) decoding. Our HESM approach lead to a considerable summarization performance boost, and when evaluated on held-out challenge data achieved a ROUGE-L of 32.77, which was the best-performing system at the top of the shared task leaderboard.[1]

1 Introduction

Summarization is a common natural language generation (NLG) task with growing recent interest (El-Kassas et al., 2021). The 1A shared task of BioNLP 2023 considers medical problem list summarization (Gao et al., 2023), where patient notes are summarized to assist medical diagnosis applications. There are several challenges faced in designing systems for this task: First, the challenge is low-resource, with only 765 examples available for training/validation. Second, a high-stake application with specialized medical terms requires systems that can deal with domain-specific terms and find relevant diagnoses from patient documents.

This paper introduces Hierarchical Ensemble of Summarization Models (HESM), an approach that is composed of two sequential ensembling layers,
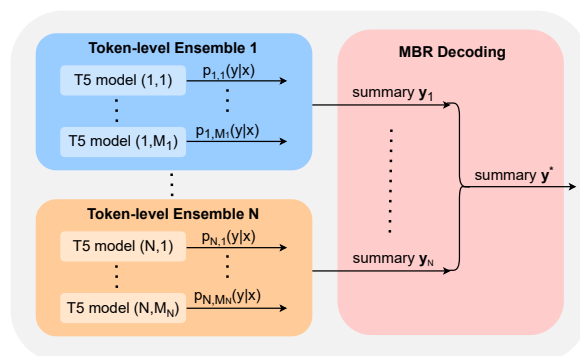

Figure 1: Hierarchical ensemble of summarization models where each individual model is a fine-tuned Clinical-T5.

of token-level ensembles, followed by Minimum Bayes Risk (MBR) decoding, as shown in Figure 1. Ensembling methods combine the predictions of various models (Fort et al., 2020) and have been effective in NLG tasks, such as in summarization (Manakul and Gales, 2020). For low-resource settings, this method allows outputs to be composed from multiple different ensemble members, which can reduce the influence of noise and eliminate spurious signals, reducing the chance of medically inaccurate summaries. We demonstrate that using HESM with Clinical-T5 models (Lehman et al., 2023) leads to systems that have a good grasp of medical knowledge, and that are able to generate outputs that are consistently closer to ground-truth summaries. Our proposed HESM method was submitted to the BioNLP problem list summarization challenge, and achieved the top position of the shared task leaderboard, out of 9 teams.

2 Background and Related Work

2.1 Existing Pre-Trained Language Models

A common approach for current NLP applications has been the pre-train and fine-tuning paradigm, where pre-trained models are fine-tuned to specific

---

[1] Our code is available at https://github.com/potsawee/hierarchical_ensemble_summ.

target tasks. The community has open-sourced a variety of pre-trained backbones of different sizes and architectures, including encoder-only such as BERT (Devlin et al., 2019), decoder-only such as GPT-2 (Radford et al., 2019), and encoder-decoder such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). Medical domain versions of language models have also been created, including ClinicalBERT (Huang et al., 2019), BioBERT (Lee et al., 2020), BioMed-RoBERTa (Gururangan et al., 2020), BioGPT (Luo et al., 2022), and Clinical-T5 (Lehman et al., 2023).

2.2 Summarization Methods

The two main summarization approaches are extractive methods, which select relevant words/phrases present in the input for the summary, and abstractive methods, which can freely generate text (even text that may not be present in the source). Previous work in medical list summarization demonstrated significant gain from adapting BART and T5 (both are abstractive models) to the medical domain and by fine-tuning them for summarization (Gao et al., 2022). For long-input summarization, a preliminary stage of selecting the most relevant input sentences before summarization was shown to be effective (Manakul and Gales, 2021). Alternatively, large language models have recently shown success in zero-shot summarization (Brown et al., 2020).

3 Hierarchical Ensemble of Summarization Models

When working with a small dataset, individual models are prone to overfit specific aspects of the data due to the limited number of training samples. By training multiple models on the same dataset, each model can potentially capture different aspects of the data that are generalizable and not prone to overfitting. Combining these diverse models together can then create a more robust and accurate prediction model (Sim et al., 2007).

Various approaches can be used to create diverse individual systems. A simple approach is to use different weights' initialization (Lakshminarayanan et al., 2017) for different seeds. Alternatively for pre-trained systems (as considered in this work), one can set different random seeds, which will influence training dropout and stochastic gradient descent batch creation, resulting in variability in the final models' weights. One can also use a form

of data *bagging*, where a different subset of the data is used to train each model (Galar et al., 2011). For example for the clinical notes, one model can be trained using only the *assessment* section of the notes, whilst another can be trained using the *assessment*+*subjective* sections.

Given an ensemble of diverse models, one may then combine them for a more robust ensemble system. A possible model combination method is weight averaging. Although weight averaging across training runs has shown success in image classification (Wortsman et al., 2022), weight averaging across different training runs is expected to work only when individual runs operate in similar weight spaces. This limits the types of combinations for weight averaging to only models with the same architecture and the same input format. As a result, we focus instead on two other methods of combination: token-level ensembling and Minimum Bayes Risk decoding (Rosti et al., 2007a,b).

Token-level Ensemble

Token-level ensemble (also known as product-of-expectations) is a technique to improve the performance of sequence-to-sequence models by combining predictions from multiple models at the token-level (Sennrich et al., 2015; Freitag et al., 2017; Malinin and Gales, 2021; Fathullah et al., 2021). Let us consider M different models, where we want to generate an output sequence, $\mathbf{y} = y_0, y_1, \ldots,$ from an input sequence $\mathbf{x}$. In the standard decoding setup, we can generate each token sequentially:

$$p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{x}, y_{<i}) \qquad (1)$$

In token ensembling, each token's probability is the average probability of the individual models:

$$p(y_i|\mathbf{x}, y_{<i}) = \frac{1}{M} \sum_m p_m(y_i|\mathbf{x}, y_{<i}) \qquad (2)$$

Minimum Bayes Risk (MBR) Decoding

Given all possible output sequences, $\mathcal{Y}$, standard decoding (inference) strategies such as beam search are used to select the sequence with the greatest likelihood:

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{Y}} \{p(\mathbf{y}|\mathbf{x})\} \qquad (3)$$

However, the above method is not well aligned with the final reward metric, $\mathcal{R}$, used to assess the quality of samples (e.g. ROUGE-L). Following MBR decoding (Kumar and Byrne, 2004; Sim

et al., 2007), we can seek to select the most *average* sample, $\mathbf{y}^* \in \mathcal{Y}$, as per our desired reward metric:

$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathcal{Y}}{\arg\max} \left\{ \mathbb{E}_{p(\tilde{\mathbf{y}}|x)}[\mathcal{R}(\mathbf{y}, \tilde{\mathbf{y}})] \right\} \qquad (4)$$

where $\mathbf{y}^*$ is expected to be the most representative of all generated samples. In practice, with only access to N sequences, $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, sourced from different model structures, there may not be available sensible or meaningful posterior distributions, $p(\mathbf{y}|\mathbf{x})$ and hence we approximate the expectation as a simple average, where each observed output $\mathbf{y}$ is taken to be equiprobable:

$$\mathbf{y}^* \approx \underset{\mathbf{y} \in \mathcal{Y}}{\arg\max} \left\{ \sum_{n=1}^{N} \mathcal{R}(\mathbf{y}, \tilde{\mathbf{y}}_n) \right\} \qquad (5)$$

The selection of $\mathbf{y}^*$ can also be viewed as a method to automatically reject the anomalous samples and thus improve overall performance. Previous work showed that MBR decoding improves machine translation (Rosti et al., 2007a,b), and the highest evaluation score is obtained when $\mathcal{R}$ matches the evaluation metric (Freitag et al., 2022). Thus, we use ROUGE-L as the reward metric $\mathcal{R}$.

We note that MBR decoding is applied at inference time, and it can be applied to any set of models regardless of their architectures or training techniques, but it is expected to be effective when there is diversity in the models' outputs.

Hierarchical Ensembling

Finally, hierarchical ensembling is a method that aims to combine the above two approaches. Multiple output sequences, $\mathcal{Y}$, can be generated by performing token ensembling over different sets of individual models. Subsequently, MBR decoding can be used to select the most representative sample from these different output sequences to give a single output sequence, $\mathbf{y}^*$. This hierarchical structure is depicted in Figure 1.

4 Experiments

4.1 Experimental Setup

Data. Training data consists of 765 progress notes along with output medical summaries, which were sourced from MIMIC-III. Due to the small amount of data available, systems were initially evaluated using 5-fold cross-validation. The test (held-out competition) data consists of 237 progress notes, where for the competition evaluation we submitted generated summaries onto an online platform

where the ROUGE-L was calculated. ROUGE-1, ROUGE-2 and ROUGE-L (Lin, 2004) were all computed during cross-fold validation.

The medical reports have three fields available: 'assessment' {A}, 'objective' {O}, and 'subjective' {S}, with word statistics shown in Table 1. We also consider different permutations by concatenating fields, separated by special tokens.

Field	{O}	{S}	{A}	Summary
#words	$304.7_{\pm 83.4}$	$85.5_{\pm 54.8}$	$33.7_{\pm 17.1}$	$10.5_{\pm 7.5}$

Table 1: Medical report and summary statistics.

Models. For abstractive summarization, we consider T5 and Clinical-T5 as the backbone. Clinical-T5[2] was initialized from scratch and pre-trained on the union of MIMIC-III and MIMIC-IV databases (Lehman et al., 2023). The models are downloaded through HuggingFace; we finetune models with teacher forcing on our training data and use beam search during inference. More details about training and inference are provided in Appendix A.

4.2 Baseline Selection

We start our investigation by comparing zero-shot, extractive, and abstractive summarization methods. Note that we provide the results of zero-shot summarization based on open-sourced large language models in Appendix B.1.

Extractive Summarization

To obtain an *empirical* upper bound, we compute ROUGE-1 between each input sentence against its ground-truth summary. The input sentences are ordered by the sections {A}, {S}, {O}.

We consider two oracle options: (1) *All-overlap*, which concatenates all input sentences where ROUGE-1 recall is positive to the generated summary; and (2) *Greedy-best*, which uses a greedy algorithm to obtain extractive sentences similar to Nallapati et al. (2017). This greedy-best method iteratively adds sentences one at a time to the generated summary, where the added sentence is the one which yields the highest ROUGE-1 (F1) score. This process is repeated until the ROUGE-1 (F1) of the generated summary does not improve. Our results in Table 2 show that even the oracle (greedy-best) approach achieves lower scores than finetuned T5 models (in Table 3).

[2]https://huggingface.co/xyla/Clinical-T5-Large.

Method	R1	R2	RL
Oracle (All-overlap)	20.37	6.41	15.44
Oracle (Greedy-best)	29.14	11.09	22.74

Table 2: Empirical upper bounds of extractive summarization methods on the training data.

Abstractive Summarization

Our first experiment is to determine the best transformer backbone for abstractive summarization. Table 3 shows T5 performance when fine-tuned using the assessment field only. We find that T5-Large significantly outperforms T5-Small, but that performance does not further scale with size with T5-Large and T5-XL performing similarly.

Pre-Trained Model	R1	R2	RL
T5-Base	26.77	10.33	24.82
T5-Large	29.56	12.01	27.88
T5-XL	29.46	12.72	27.58
Clinical-T5-Base	26.62	12.11	24.96
Clinical-T5-Large	**32.22**	**14.30**	**30.15**

Table 3: ROUGE-scores of various T5/Clinical-T5 models on cross-validation, with inputs being {A} only.

We further find that domain adaptation can lead to an additional boost, with Clinical-T5-Large showing ROUGE scores 2% higher than T5-Large. Domain adaptation, however, was not helpful for our low-capacity model, with Clinical-T5-Base performing similarly to T5-Base. We, therefore, use **Clinical-T5-Large**, the best-performing system during cross-validation, as our backbone transformer for all further systems.

The next experiments consider which input fields are most useful for generating the summary. Table 4 shows that the assessment field, {A}, contains the key information for the patient's problem summary with ROUGE scores below 20 when any other field is used alone. We further observe better performance when {A} is augmented with {S}.

Inputs	R1	R2	RL
∅	9.61	2.93	9.24
{O}	14.09	4.99	13.27
{S}	17.96	7.00	16.81
{A}	32.22	14.30	30.02
{A}+{S}	33.46	15.07	31.03

Table 4: Comparison of Clinical-T5-Large performance when using different inputs on training data (using cross-validation). The empty input baseline ∅ is trained to generate summaries without the input report.

4.3 Ensemble Methods

To maximize the performance, we apply ensemble methods to fine-tuned Clinical-T5-Large models. For the simplicity of notation, we use θ_A and θ_{AS} to denote the system where the input is {A} and {A}+{S}, respectively. We train nine θ_A individual models where all models are initialized from Clinical-T5-Large weights, and have the same training hyperparameters except random seeds for data batching. In Table 5, we compare different methods for combining nine θ_A models and the results show that: 1) weight averaging results in a slightly worse system; 2) both token-level ensemble and MBR decoding yield better performance than single models. In addition, we observe a similar trend when combining θ_{AS} models as shown in Table 6.

Method		ROUGE-L	
	F_1	Prec	Rec
Individual	$29.84_{\pm 0.69}$	$40.11_{\pm 1.40}$	$27.68_{\pm 0.76}$
Weight Avg.	29.39	39.68	27.50
Tok. Ensemble	30.50	41.09	28.37
MBR Decoding	30.72	40.96	28.91

Table 5: ROUGE-L on the test data. This table compares combination methods of nine θ_A models.

Method		ROUGE-L	
	F_1	Prec	Rec
Individual	$29.44_{\pm 0.45}$	$37.57_{\pm 0.69}$	$28.33_{\pm 0.65}$
Weight Avg.	28.00	35.65	27.16
Tok. Ensemble	30.04	36.35	29.96
MBR Decoding	30.30	38.39	28.92

Table 6: ROUGE-L on the test data. This table compares combination methods of nine θ_{AS} models.

Hierarchical Ensemble. We explore combining θ_A and θ_{AS} models in a token-level ensemble followed by MBR decoding to form a hierarchical ensemble. Based on nine θ_A models and nine θ_{AS} models, Table 7 provides the results of hierarchical combination in different setups.

The first block shows the performance when combining one θ_A and one θ_{AS} in a token-level ensemble, followed by an MBR combination stage over 9 of these ensembles. Similarly, the second block shows the performance when combining three θ_A and three θ_{AS} each in a token-level ensemble fashion followed by an MBR decoding stage over 3 of these ensembles.

519

Name	Ensemble		ROUGE-L		
	Token	MBR	F_1	Prec	Rec
$\theta_A+\theta_{AS}$	(1, 1)	✗	$31.17_{\pm0.67}$	$39.51_{\pm1.30}$	$29.66_{\pm1.02}$
HESM	(1, 1)	9	32.31	41.16	30.16
$\theta_A+\theta_{AS}$	(3, 3)	✗	$31.50_{\pm0.42}$	$39.74_{\pm0.79}$	$29.97_{\pm0.57}$
HESM	(3, 3)	3	31.87	39.63	30.24
	(3, 3)	9	31.88	40.07	30.17

Table 7: ROUGE-L of HESM on the test data. (a, b) denotes token-level ensemble consisting of $a\theta_A$ models and $b\theta_{AS}$ models. MBR=c denotes the outputs of c token-level ensembles combined using MBR decoding. For HESM(3,3) w/ MBR=3, ensembles with non-overlap members are chosen.

4.4 Evaluation System

This section discusses the specific nature of our HESM systems submitted to the shared task. Given the flexibility in an MBR combination, members of HESMs are not limited to token-level ensembles. Hence, during the competition we made use of previously submitted systems to build the final HESM. As a result, our HESM consists of six systems: best-performing θ_A; weight averaging of $3\theta_A$; token-level ensemble of $3\theta_A$ with $\mathcal{L}_{RL}$; $2\times$token-level ensemble of $9\theta_A$; and token-level ensemble of $9\theta_{AS}$. The results of the HESM's members are provided in Table 11 in the appendix.

We further consider combining this HESM with the token-level ensembles of $3\theta_A+3\theta_{AS}$ investigated in Section 4.3. The first ensemble (v1) is obtained by selecting three θ_A and three θ_{AS} (out of the nine θ_A and nine θ_{AS}) with the lowest cross-entropy training losses. The second ensemble (v2) is obtained by training variants of the three θ_A and three θ_{AS} in the first ensemble using different hyperparameters to increase diversity. The results of these two token-level ensembles are reported in Table 8.

Ultimately, we combine the above HESM with these two ensembles using MBR decoding, and this combined system can be viewed as a higher level of HESM as it consists of HESM as a member of the MBR combination. This final combination sets the state-of-the-art performance of the task, achieving the ROUGE-L score of 32.77.

System	ROUGE-L		
	F_1	Prec	Rec
HESM	31.86	43.52	28.90
TokEns($3\theta_A+3\theta_{AS}$)-v1	32.03	41.01	30.16
TokEns($3\theta_A+3\theta_{AS}$)-v2	32.19	39.59	30.88
+ MBR Combination[†]	**32.77**	41.69	30.51

Table 8: ROUGE-L on the test data. [†]This system attains the top position on the shared task leaderboard.

5 Conclusions

In low-resource and medical-domain summarization, our work has demonstrated that abstractive summarization outperforms extractive and zero-shot methods. Furthermore, both token-level ensemble and MBR decoding improve the overall performance. Our HESM, which utilizes both ensembling techniques, achieves state-of-the-art performance with the highest ROUGE-L score in the BioNLP 2023's shared task 1A leaderboard.

6 Limitations

The limitations of this work are mainly that there is a small amount of data available for inference to test the models. ROUGE-L is used as an assessment metric and n-gram overlap metrics are notably not optimal for abstractive summarization assessment (Zhang* et al., 2020; Deutsch, 2022).

7 Ethics Statement

The study used de-identified health data to develop a system that overcomes biases in medical decision-making. However, social biases in language models need to be addressed to ensure fairness in model training. Therefore, before deploying any pretrained language model, fairness audits are necessary to ensure an ethical and trustworthy model for all stakeholders. Note, doctors should not rely on automated summarization systems for diagnoses in the interest of patient care.

Acknowledgements

References

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceed-*

ings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Deutsch. 2022. *Methods for Text Summarization Evaluation*. Ph.D. thesis, University of Pennsylvania.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Yassir Fathullah, Mark J.F. Gales, and Andrey Malinin. 2021. Ensemble distillation approaches for grammatical error correction. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2745–2749.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2020. Deep ensembles: A loss landscape perspective.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. 2011. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484.

Yanjun Gao, Dmitriy Dligach, Timothy Miller, Dongfang Xu, Matthew M. M. Churpek, and Majid Afshar. 2022. Summarizing patients' problems from hospital progress notes using pre-trained sequence-to-sequence models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2979–2991, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yanjun Gao, Dmitry Dligach, Timothy Miller, Matthew M. Churpek, and Majid Afshar. 2023. Overview of the problem list summarization (probsum) 2023 shared task on summarizing patients' active diagnoses and problems from electronic health record progress notes. In *Proceedings of the 22nd Workshop on Biomedical Language Processing*, Toronto, Canada. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv:1904.05342*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Eric Lehman, Evan Hernandez, Diwakar Mahajan, Jonas Wulff, Micah J Smith, Zachary Ziegler, Daniel Nadler, Peter Szolovits, Alistair Johnson, and Emily Alsentzer. 2023. Do we still need clinical language models? *arXiv preprint arXiv:2302.08091*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6). Bbac409.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Potsawee Manakul and Mark Gales. 2020. CUED_SPEECH at TREC 2020 podcast summarisation track. *arXiv preprint arXiv:2012.02535*.

Potsawee Manakul and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041, Online. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie Dorr. 2007a. Combining outputs from multiple machine translation systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 228–235, Rochester, New York. Association for Computational Linguistics.

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. 2007b. Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 312–319, Prague, Czech Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

K. C. Sim, W. J. Byrne, M. J. F. Gales, H. Sahbi, and P. C. Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, pages IV–105–IV–108.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A More details about experiments

A.1 Inference Hyperparameters

num_beams = 4, length_penalty = 0.6, min_length = 5, max_length = 256, no_repeat_ngram_size = 4.

A.2 RL training

We follow Paulus et al. (2018) in using reinforcement learning (RL) based loss:

$$\mathcal{L}_{\text{RL}} = (\mathcal{R}(\bar{\mathbf{y}}, \mathbf{y}) - \mathcal{R}(\hat{\mathbf{y}}, \mathbf{y})) \log P(\hat{\mathbf{y}}|\mathbf{x}) \quad (6)$$

where $\bar{\mathbf{y}}$ is the sequence obtained by greedy search, $\hat{\mathbf{y}}$ is the sequence obtained by sampling, and $\hat{\mathbf{y}}$ is the ground-truth sequence. To improve the stability of training, we initialize the model using the weights from the maximum likelihood training (cross-entropy loss), and we use a combined loss: $\mathcal{L} = \gamma \mathcal{L}_{\text{RL}} + (1 - \gamma) \mathcal{L}_{\text{ML}}$ where $\gamma = 0.9$ and $\mathcal{L}_{\text{ML}}$ is the standard cross entropy loss. The results are provided in Table 11, showing that a marginal gain can be achieved from RL training.

B Additional Results

B.1 Zero-shot Summarization

Since state-of-the-art LLMs such as GPT-3 or Chat-GPT are only available via API services, using them would violate the MIMIC data use agreement. Instead, we use open-source LLMs. We use the following text input to large language models (LLMs),

```
{prompt}:
{clinical_note}
```

where `clinical_note` is the *assessment* section, and we consider two prompts:

- P1: *Summarize these clinical notes.*

- P2: *Give a one or two word summary for these clinical notes.*

We use open-source LLMs including OPT-IML (Iyer et al., 2022), GPT-J (Wang and Komatsuzaki, 2021), and GPT-NeoX (Black et al., 2022). We provide the results on training data in Table 9. The poor performance could be attributed to the small size of LLMs, and larger systems such as GPT-3 or ChatGPT could potentially perform much better.

LLM	prompt	R1	R2	RL
OPT-IML-1.3B	P1	4.97	0.80	4.37
	P2	4.46	0.65	4.05
OPT-IML-30B	P1	2.76	0.44	2.51
	P2	2.07	0.36	1.96
GPT-J-6B	P1	4.13	0.58	3.68
	P2	4.66	0.73	4.29
GPT-NeoX-20B	P1	2.41	0.38	2.21
	P2	3.04	0.59	2.79

Table 9: Zero-shot Summarization performance on training data of LLMs with different user prompts.

B.2 More Analysis

Table 10 shows our post-evaluation studies on the performance using different input fields, and the results suggest that it is possible to improve the performance further by using {A}+{S}+{O} in addition to {A} and {A}+{S} as the input.

B.3 Submitted Systems

In Table 11, we present other approaches that were submitted to the shared task, including model weight averaging, and RL-based training. These models also formed components of the final HESM model submitted.

Inputs	R1	R2	RL
{A}	32.22	14.30	30.02
{A}+{O}	32.50	13.81	30.31
{A}+{S}	33.46	15.07	31.03
{A}+{S}+{O}	33.80	15.38	31.28

Table 10: Comparison of Clinical-T5-large performance when using different inputs on training data (using cross-validation).

System	ROUGE-L		
	F_1	Prec	Rec
Weight Avg. of $3\theta_A$	30.26	42.51	27.31
TokEns $3\theta_A$ w/ $\mathcal{L}_{RL}^{\dagger}$	30.40	43.78	27.10
Best-performing θ_A	30.56	39.97	28.95
TokEns $9\theta_A$-v1	30.74	42.14	27.93
TokEns $9\theta_A$-v2	30.50	41.09	28.37
TokEns $9\theta_{AS}$	30.04	36.35	29.96

Table 11: ROUGE-L scores on test data of the members of HESM. $^{\dagger}\mathcal{L}_{RL}$ is described in Appendix A.2.

C Post-evaluation Ablation Study

The results in Table 8 found that a higher-level Hierarchical Ensemble (HESM) model had the best performance. This model performs MBR decoding over the output from an existing shallower HESM model (we will refer to as HESM-shallow) and $2\times$token-level ensemble of $3\theta_A+3\theta_{AS}$. Table 12 explores the impact on performance with unpacking the token level-ensemble systems and performing MBR decoding over all individual systems. The system labelled with *unpack-1* performs MBR decoding over the 6 ensemble systems that form HESM-shallow and the 12 individual systems used to make the two token level ensemble systems; i.e. MBR decoding is performed over **18** system output sequences. The *unpack-2* system considers further unpacking the 6 ensemble systems used for HESM-shallow, such that MBR decoding is now performed over a total of **34** unique individual systems.

System	ROUGE-L		
	F_1	Prec	Rec
HESM (final)	32.77	41.69	30.51
HESM-unpack-1	32.38	42.98	29.91
HESM-unpack-2	32.26	43.18	29.62

Table 12: ROUGE-L scores on the test data. This table considers the impact of performing MBR decoding on the individual systems after unpacking token-level ensemble systems used as components for the final higher-level HESM model submitted in the competition in Table 8.