UoG Siephers at "Discharge Me!": Exploring Ways to Generate Synthetic Patient Notes From Multi-Part Electronic Health Records

Erlend Frayling, Jake Lever and **Graham McDonald**
University of Glasgow
Scotland, UK

firstname.lastname@glasgow.ac.uk

Abstract

This paper presents the UoG Siephers team participation at the Discharge Me! Shared Task on Streamlining Discharge Documentation. For our participation, we investigate appropriately selecting and encoding specific sections of Electronic Health Records (EHR) as input data for sequence-to-sequence models, to generate the *discharge instructions* and *brief hospital course* sections of a patient's EHR. We found that, despite the large volume of disparate information that is often available in EHRs, selectively choosing an appropriate EHR section for training and prompting sequence-to-sequence models resulted in improved generative quality. In particular, we found that using only the *history of present illness* section of an EHR as input often led to better performance than using multiple EHR sections.

1 Introduction

In the clinical domain, writing notes about patients' health, diagnoses, and treatments is a necessary part of the patient healthcare journey, but it is also time consuming (Weiner and Biondich, 2006; Sinsky et al., 2016). The time spent by essential care staff, such as doctors and nurses, writing the notes in Electronic Health Records (EHRs) could be time better spent performing important clinical duties.

The *Discharge Me!* BioNLP ACL'24 Shared Task on Streamlining Discharge Documentation challenged participants to produce a system that can automatically generate: discharge instructions, which contain detailed guidelines provided to a patient upon their discharge from hospital; and Brief Hospital Courses, which summarise the key events, treatments and progress for a patient during their hospital stay (Xu et al., 2024). Discharge Me! participants were provided a dataset curated from the MIMIC-IV database (Johnson et al., 2023), which contains de-identified patients' EHRs.

EHRs are complex collections of, often long and disparate, reports about a patient's stay in hospital, including reports on patient demographics, medical history, laboratory tests and results, instructions for the patient and many more sections. In this work, we investigate several ways of appropriately selecting and encoding specific sections of EHRs as input data for sequence-to-sequence (seq2seq) models to generate the two target sections of the Discharge Me! shared task, i.e., the *discharge instruction* and the *brief hospital course*. In particular, in this work we investigate the following three research questions that guide our experimentation:

RQ1: What is the effect of using different sections of EHRs as training data for seq2seq models?

RQ2: Can a model that uses multiple EHR sections as input achieve better performance than models trained on single sections of EHRs?

RQ3: When concatenating multiple EHR sections as input, is it better to concatenate lexically, or concatenate embeddings post-encoding?

2 Related Work

Most relevant to our work is that of Hartman and Campion (2022), who employed various encoder-decoder models with different pre-trained checkpoints (Rothe et al., 2020) to generate a brief hospital course. Hartman and Campion attempted to summarise EHR records as short day-by-day summaries so that the EHR summaries would fit within the context limit of seq2seq encoder-decoder models. In our work, instead of summarising the input data to fit the context limit of an encoder-decoder model, we experiment with selectively choosing individual subsections of the EHR records to train seq2seq models.

Pal *et al.* (Pal et al., 2023) used the *nursing report* section of EHRs to generate a variety of EHR sections, such as the *history of present illness* and *discharge instructions*. The authors showed that

Proceedings of the 23rd Workshop on Biomedical Language Processing, pages 712–718

seq2seq models, such as T5 and BART, can be effective for this task (Raffel et al., 2020; Lewis et al., 2019). Differently from Pal *et al.*, we explore using multiple sections of the EHR as input data, and ways to combine EHR sections as input.

Finally, the work of Liu et al. (2022) used the discharge instructions of historic patients, who had similar symptoms to a new patient, to write the new patient's discharge instruction. They used graph-based reasoning to generate the new discharge instruction based on the historic patients' instructions. Differently, we focus on using information that is entirely available in the patients own record and do not rely on the information of other patients.

3 Methods

In this section we describe our approaches for generating discharge instruction and brief hospital course sections of EHRs. Using different pre-trained encoder and/or decoder models within seq2seq models has been shown to be an effective way to adapt such models for different tasks (Rothe et al., 2020; Hartman and Campion, 2022). Therefore, in this work, we investigate three approaches for constructing the input data for seq2seq models, such that we can use the models' limited context effectively to model the EHRs sections. For each of our approaches, we deploy encoder-decoder models following the work of Hartman and Campion (Hartman and Campion, 2022).

3.1 Separate Text Sections

Our first approach uses individual EHR sections as the input to the seq2seq model. By using a specific self-contained section, we ensure that the training data is a focused and coherent report about the patient's medical history. In our experiments, we compare the effectiveness of two separate EHR sections, namely: History of Present Illness (HPI), which contains information about patients' stays in hospital; Radiology Report (RR), which contains information about patients' radiology exams.

We choose to evaluate the HPI and RR sections since the HPI sections encompass a lot of information that is also discussed in other sections of EHRs. Therefore, HPI sections can act as a general overview of the patient's condition, their reason for visiting the hospital, and their care plan. Differently, the RR section details specific observations about the physical condition of the patient. Indeed, there is little intersection between the information

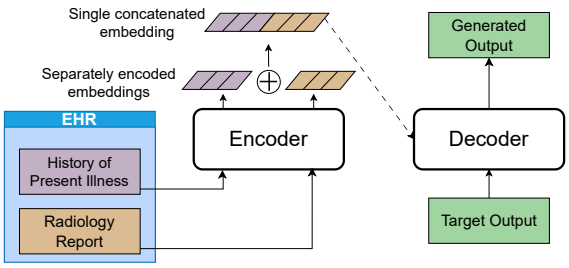

Figure 1: Two EHR sections (purple and orange) are passed to the encoder separately, then their separate embedded representations, of corresponding colour, are concatenated before being passed to the decoder model.

that is contained in the HPI and RR sections. Therefore, evaluating the sections separately can provide valuable insights about what kinds of information are most useful for automatically generating discharge instructions and brief hospital courses.

3.2 Concatenating Text Sections

Secondly, we consider that multiple EHR sections may contain information that is essential for generating a correct discharge instruction or brief hospital course. In such a case, the approach presented in Section 3.1 would not be able to model all of the required information, due to the limited context of seq2seq models. Therefore, in this approach, we concatenate the text of the HPI and RR sections and use an encoder model that accepts longer context, i.e., Longformer (Beltagy et al., 2020).

3.3 Concatenating Embedded Sections

Finally, instead of concatenating the *text* of the different EHR sections, as described in Section 3.2, in this approach we encode the HPI and RR sections separately and then concatenate the *encoded* sections, before passing the concatenated encodings to the cross attention layers of the decoder model. This approach is inspired by how the reader component of Fusion-in-Decoders (Izacard and Grave, 2020) performs question answering tasks with multiple retrieved context documents. We therefore refer to this approach as our fusion approach.

Figure 1 illustrates our fusion approach. In this approach, a model with a shorter context limit can be used, though encoding the multiple sections *separately* increases computation time linearly. Moreover, the sections to be encoded are distinct, separate reports. Concatenating the sections into a single long passage to encode, such as in Section 3.2, may result in a low-quality embedded repre-

sentation that cannot capture the diversity of the different textually concatenated reports. In this approach, by encoding EHR sections separately, the contextual separation is retained for each section and we hypothesise that this may improve performance in the overall seq2seq generation task.

4 Experimental Setup

In this section, we present the experimental setup to investigate the three research questions that we presented in Section 1.

4.1 Dataset

The dataset for the task was provided by the organisers of the Discharge Me! shared task and is curated from the MIMIC-IV database. The data can be downloaded from Physionet.[1] The dataset is split into a training set (68,785 samples), a validation set (14719 samples), a phase 1 testing set (14702 samples) and a phase 2 testing set (10962 samples). Each sample corresponds to an emergency department admission with an associated discharge summary, which contains many different reports on a patient's stay in hospital. Each sample also contains at least one RR. Finally, each sample also contains a discharge instruction and a brief hospital course section, which are the two target sections to be generated. The dataset includes gold standard discharge instruction and brief hospital course sections for each of the training, validation, phase 1 testing and phase 2 testing sets.

For our approaches described in Section 3, we extract the HPI section and the most recent RR section (some samples contain more than one RR section) for each sample in the dataset using Python Regular Expressions. From early exploratory work, we discovered that, for the models that we evaluate, using a large number of samples for training offers little performance improvements compared to using a smaller subset of the data. We therefore use a subset dataset of 5000 random samples from the training set, and 1000 random samples from the validation set to train our chosen models.

4.2 Models

We now provide a description of the different models and model architectures that we deploy in our experiments. In all cases we train two versions of each model. One version is trained to generate the target discharge instruction, while the other is trained to generate the target brief hospital course. In all cases, we train the models using the HPI and/or RR sections to generate the target sections.

Firstly, to investigate the approach presented in Section 3.1, we evaluate several encoder-decoder seq2seq models that are trained on a single input, either HPI or RR, and leverage pre-trained checkpoints following Rothe *et al.* (Rothe et al., 2020). We deploy three encoder models: a RoBERTa encoder (Liu et al., 2019), since it was found to be the best performing by Rothe *et al.* (Rothe et al., 2020); the ClinicalBERT encoder (Alsentzer et al., 2019), as it is pre-trained on MIMIC data; and the BERT encoder (Devlin et al., 2018) as an appropriate baseline. We deploy the same decoder, GPT-2 (Radford et al., 2019), in all instances. Additionally, we also deploy a base-size T5 model (Raffel et al., 2020) since it has been shown to be effective for seq2seq tasks. Our participation in the Discharge Me! shared task investigated the effectiveness of different encoder-decoder models, however for completeness we deploy two decoder-only models, namely GPT-2 (Radford et al., 2019) and Llama 3 8B (Meta, 2024). We train the decoder-only models by passing the input and target sections as one string, where the sections are separated by three newline characters. At inference time the models are passed only the input data and newline characters.

For our approach that we described in Section 3.2, we deploy an encoder-decoder model with a pre-trained Longformer encoder model (Beltagy et al., 2020) and GPT-2 decoder. The Longformer model has a context window of up to 4096 tokens, ensuring that for EHR samples in our dataset the HPI and RR sections can be concatenated before encoding, as described in Section 3.2. We concatenate the sections as separate paragraphs by joining the sections with connecting newline characters.

Finally, for our fusion approach, presented in Section 3.3, we deploy two encoder-decoder models. Both models use a RoBERTa encoder and GPT-2 decoders. One model uses the base size GPT-2 decoder, the other model uses GPT-2 large. We refer to these models as Fusion-roBERTa-GPT2 and Fusion-roBERTa-GPT2-large respectively.[2]

For all of our models, we perform 15 runs of hyperparameter tuning using Optuna (Akiba

[1] https://physionet.org/

[2] We also submitted BERT-GPT2 and ClinicalBERT-GPT2 fusion runs to the Discharge Me! leader board. However, the official evaluation script was not able to generate results for us to evaluate these models.

Model	Overall Score	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	Meteor	AlignScore	MEDCON
Encoder-Decoder Architectures									
T5 (HPI)	**0.191**	0.017	**0.341**	**0.108**	**0.209**	**0.268**	0.247	0.143	**0.193**
T5 (RR)	0.079	0.001	0.128	0.008	0.080	0.130	0.073	**0.157**	0.054
BERT-GPT2 (HPI)	0.144	0.012	0.258	0.045	0.124	0.245	0.242	0.113	0.114
BERT-GPT2 (RR)	0.156	0.011	0.294	0.055	0.157	0.253	0.244	0.105	0.128
roBERTa-GPT2 (HPI)	0.143	0.009	0.250	0.025	0.135	0.251	0.239	0.107	0.124
roBERTa-GPT2 (RR)	0.110	0.005	0.183	0.010	0.092	0.198	0.188	0.119	0.083
ClinicaBERT-GPT2 (HPI)	0.143	0.011	0.254	0.020	0.131	0.252	0.239	0.108	0.124
ClinicalBERT-GPT2 (RR)	0.149	0.012	0.272	0.046	0.144	0.252	0.240	0.106	0123
LongFormer-GPT2	0.152	0.013	0.278	0.030	0.153	0.255	0.244	0.110	0.135
Fusion-roBERTa-GPT2	0.148	0.011	0.264	0.029	0.137	0.255	0.243	0.113	0.130
Fusion-roBERTa-GPT2-large	0.159	**0.039**	0.222	0.042	0.146	0.251	**0.266**	0.134	0.169
Decoder-only Models									
GPT-2 (HPI)	0.153	0.009	0.284	0.035	0.139	0.241	0.206	0.167	0.151
GPT-2 (RR)	0.128	0.011	0.160	0.021	0.101	0.232	0.212	0.158	0.131
Llama 3 (HPI)	0.196	0.028	0.350	0.091	0.180	0.300	0.218	0.172	0.230
Llama 3 (RR)	0.168	0.016	0.322	0.072	0.160	0.264	0.195	0.151	0.167

Table 1: Results for our different methods evaluated by the official Discharge Me! submission system. **Bold text** indicates the best scoring encoder-decoder results. Underlined text indicates the best scoring decoder-only results.

et al., 2019), searching learning rate (1e-6 to 1e-3), weight decay (1e-4 to 1e-2), and number of epochs (1 to 9). We optimise for evaluation loss and use the best hyperparameter configuration to train a final model that is used in evaluation, all using 3 NVIDIA RTX A6000 GPUs. To fine-tune the Llama 3 model we use gradient accumulation (Goodfellow et al., 2016; Bengio, 2012), processing batches of 4 and accumulating to batches of 8, 8 being the batch size we use to train all of our other models. We also use Quantised Low Rank Adaptation to fine-tune the Llama 3 model (Dettmers et al., 2024)

4.3 Evaluation Metrics

The two generated target texts of each model are evaluated independently against their corresponding gold standard texts using a variety of text-based similarity metrics and factual correctness metrics. The metrics used for evaluation are: BLEU-4 (Papineni et al., 2002); the ROUGE metrics including ROUGE-1, ROUGE-2 and ROUGE-L (Lin, 2004); BertScore; Meteor (Banerjee and Lavie, 2005); AlignScore (Zha et al., 2023), and MEDCON (Yim et al., 2023). Each of the eight metric scores for each of the two generated datasets are then averaged to get combined score for each metric, and then finally all eight scores are average again to produce a single *overall score*.

5 Results

This section describes our findings relating to the research questions presented in Section 1. Table 1 provides the results our models achieved when submitted to Discharge Me! leaderboard (Xu et al., 2024). Overall for the encoder-decoder models, a T5 model trained on HPI sections of patient EHRs was the best performing model, achieving 0.191 Overall Score, with the next best approach BERT-GPT2(RR) achieving 0.156 Overall Score. Additionally, the Llama 3 decoder-only model achieves competitive performance with the T5 model when using the HPI sections of EHRs. This is, arguably, to be expected given the much larger size, and recency of the model. Furthermore, both decoder-only models, Llama 3 and GPT-2, perform better when using the HPI as input rather than the RR section. This is in line with our findings for encoder-decoder models.

Concerning RQ1, the T5 and RoBERTa-GPT2 models perform better when trained on the HPI input. On the other hand, the BERT-GPT2 model and ClinicalBERT-GPT2 model perform better when trained with RR input. However, the performance increases that are obtained from training on HPI data are notably greater than any performance improvements that are obtained from training on RR data. The T5(HPI) model shows 141% improvement in Overall Score compared to the T5(RR) model, whereas the BERT-GPT2(HPI) model resulted in only a 7% Overall Score drop compared to its BERT-GPT2(RR) counterpart. Similarly, roBERTa-GPT2(HPI) achieves a 29% improvement over roBERTa-GPT2(RR), while there is only a 4% drop in Overall Score between ClinicalBERT-GPT2(RR) and ClinicalBERT-GPT2(HPI). Answering RQ1, we find that when training on individual record sections, training on HPI most often

leads to better performance compared to models trained on RR. Indeed, the gains in Overall Score from training on HPI compared to RR are notably greater.

In answering RQ2, we find that training seq2seq models on multiple concatenated sections of EHR with models does not outperform models trained on a single input section of a record. Our best performing concatenation model, LongFormer-GPT-2, outperforms several models trained on single EHR sections. However, both BERT-GPT2(RR) and T5(HPI) both outperform the Longformer-GPT2 model. This indicates that choosing a single input section for the *right* model can outperform a model that has access to both sections of the data. Specifically, we see that the Longformer-GPT2 outperforms the BERT-GPT2(RR) model by a small margin in BERTScore and Meteor. However, the two models perform very similarly, indicating that the additional information in HPI that was available to the Longformer-GPT2 model did not improve its performance markedly. Thus, training on additional information is not always beneficial.

Regarding RQ3, we find that concatenating the different sections of EHR records lexically, and then using an encoder with a larger context window is a more effective method for this task than encoding the different sections separately as proposed in Section 3.3. Neither of our fusion models beat the LongFormer-GPT2 model in overall score, despite Fusion-roBERTa-GPT2-large using a larger decoder model. Considering this in the context of the findings of our first and second research question, this may indicate that the decoder model is not able to utilise the separately embedded records sections as effectively as it is able to understand embeddings of a single section of the report. Replacing the decoder with a larger model does improve performance, but still the performance is worse than a T5-base model trained on the single HPI section.

6 Qualitative Analysis

In this section we investigate the generated record quality for the best performing seq2seq model, T5(HPI). We analyse the ten highest and ten lowest scoring generated discharge instructions and brief hospital course, in terms of their ROUGE-1 scores.

In the highest scoring generated EHR sections, the core ailments of the patients are correctly described. In the generated discharge instructions, the recommended followup treatment is often inaccurate but the structure of the instructions, which all contain subheadings (e.g. "why you were in hospital", "what you should do after leaving"), are usually correctly generated and match the target texts. This improves the overall quality of the generated discharge instructions. The generated brief hospital courses match most of the text of their corresponding target texts exactly. However, they deviate towards the end of the text often adding extended information that is still topically relevant, but not actually part of the true target text.

Inspecting the lowest scoring generated samples, we find common problems with the generation process for both the discharge instructions and brief hospital courses. While our models are effective at writing structured discharge instructions with specific sub headings, and brief hospital courses that contain a verbose description of the patient's problems, the effectiveness of the generation degrades when the target texts are not in line with these formats. For example, when the target discharge instruction is a short single-line note, such as instructions about avoiding a certain kind of food, or a reminder for the patient to weigh themselves, the models attempts to generate a long instruction with many unnecessary subsections. Similarly, our model will attempt to generate a verbose brief hospital course, even when the true target is a list of vital-sign readings. Uniquely to the discharge instruction generation, we find that several of the target sections are written in Spanish. In such cases, our model still attempts to generate English text, as the input section is always written in English.

7 Conclusion

To conclude, we have found that training a seq2seq model to generate discharge instructions and brief hospital courses using single sections from Electronic Health Records (EHR) as input, outperformed models trained using multiple sections of EHR as input. Moreover, choosing which single section to use as input is an important factor that depends on the chosen seq2seq model and that generally, some sections can expect to provide reasonable performance overall compared to others.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) grant number EP/X018237/1.

8 Limitations

We note that there are many potential extension to our experiments that could provide additional valuable insights beyond the scope of this work. Firstly, in our work we use only a GPT-2 decoder in all our encoder-decoder models, while in Table 1 we find that a Llama 3 decoder-only model outperforms the GPT-2 decoder-only model. Therefore, we could, for example, evaluate Llama 3 as the decoder in an encoder-decoder architecture. Moreover, evaluating different sizes of decoder models would also bring additional insights. For example, the results for the Fusion-roBERTa-GPT2-large model in Table 1 show that using a larger variant of GPT-2 decoder in the encoder-decoder architecture improves overall performance.

Secondly, in our work we only investigate using two sections of EHRs, namely the History of Present Illness section and the Radiology Report. Though ultimately we found using one of these two sections to train a model was more effective than combining both sections as input, further research to explore the use of other sections of the EHR poses interesting questions. For example, are there other sections that are better to use as input than the ones we have chosen to use? Moreover, concerning the approaches described in Sections 3.2 and 3.3, how does increasing the number of EHR sections that are concatenated as input affect performance?

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Vince Hartman and Thomas R Campion. 2022. A day-to-day approach for automating the hospital course section of the discharge summary. *AMIA Summits on Translational Science Proceedings*, 2022:216.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Fenglin Liu, Bang Yang, Chenyu You, Xian Wu, Shen Ge, Zhangdaihong Liu, Xu Sun, Yang Yang, and David Clifton. 2022. Retrieve, reason, and refine: Generating accurate and faithful patient instructions. *Advances in Neural Information Processing Systems*, 35:18864–18877.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Meta. 2024. Introductng meta llama 3: The most capable openly available llm to date.

Koyena Pal, Seyed Ali Bahrainian, Laura Mercurio, and Carsten Eickhoff. 2023. Neural summarization of electronic health records. *CoRR*, abs/2305.15222.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

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. *Journal of machine learning research*, 21(140):1–67.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Christine Sinsky, Lacey Colligan, Ling Li, Mirela Prgomet, Sam Reynolds, Lindsey Goeders, Johanna Westbrook, Michael Tutty, and George Blike. 2016. Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties. *Annals of internal medicine*, 165(11):753–760.

Michael Weiner and Paul Biondich. 2006. The influence of information technology on patient-physician relationships. *Journal of general internal medicine*, 21(Suppl 1):35–39.

Justin Xu, Andrew Johnston, Zhihong Chen, Louis Blankemeier, Maya Varma, Curtis Langlotz, and Jean-Benoit Delbrouck. 2024. Overview of the first shared task on clinical text generation: Rrg24 and "discharge me!". In *The 23rd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, Bangkok, Thailand. Association for Computational Linguistics.

Wen-wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. 2023. Aci-bench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Scientific Data*, 10(1):586.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.