Generating Medical Reports from Patient-Doctor Conversations using Sequence-to-Sequence Models

**Seppo Enarvi[1], Marilisa Amoia[1], Miguel Del-Agua Teba[1], Brian Delaney[1],
Frank Diehl[1], Guido Gallopyn[1], Stefan Hahn[1], Kristina Harris[1], Liam McGrath[1],
Yue Pan[1], Joel Pinto[1], Luca Rubini[1], Miguel Ruiz[1], Gagandeep Singh[1],
Fabian Stemmer[1], Weiyi Sun[1], Paul Vozila[1], Thomas Lin[2], Ranjani Ramamurthy[2]**

[1]Nuance Communications, 1 Wayside Road, Burlington, MA 01803
firstname.lastname@nuance.com
[2]Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
{tlin,ranjanir}@microsoft.com

Abstract

We discuss automatic creation of medical reports from ASR-generated patient-doctor conversational transcripts using an end-to-end neural summarization approach. We explore both recurrent neural network (RNN) and Transformer-based sequence-to-sequence architectures for summarizing medical conversations. We have incorporated enhancements to these architectures, such as the pointer-generator network that facilitates copying parts of the conversations to the reports, and a hierarchical RNN encoder that makes RNN training three times faster with long inputs. A comparison of the relative improvements from the different model architectures over an oracle extractive baseline is provided on a dataset of 800k orthopedic encounters. Consistent with observations in literature for machine translation and related tasks, we find the Transformer models outperform RNN in accuracy, while taking less than half the time to train. Significantly large wins over a strong oracle baseline indicate that sequence-to-sequence modeling is a promising approach for automatic generation of medical reports, in the presence of data at scale.

1 Introduction

There has been an increase in medical documentation requirements over the years owing to increased regulatory requirements, compliance for insurance reimbursement, caution over litigation risk, and more recently towards increased patient participation. According to a study on 57 U.S. physicians, for every hour with a patient, a physician takes an additional hour of personal time doing clerical work (Sinsky et al., 2016). Increased documentation burden has been identified as one of the main contributing factors for physician burnout (Wright and Katz, 2018). In another, larger study, U.S. physicians who used electronic health records (EHRs) or computerized physician order entry (CPOE) were found to be less satisfied with the time spent on administrative work (Shanafelt et al., 2016).

Increased physician burnout not only affects the health and well-being of the physicians, it can also lead to increased medical errors, increased job turnover, reduced productivity, and reduced quality of patient care (Panagioti et al., 2017). Factors related to physician burnout and its consequences have been studied in detail in the literature (Patel et al., 2018b).

Use of automatic speech recognition (ASR) to dictate medical documentation has contributed significantly to the efficiency of physicians in creating narrative reports (Payne et al., 2018). However the content of the report has already been discussed with the patient during the encounter. Medication list and orders entered into the EHRs are also discussed with the patient. In other words, creation of medical documentation by the physician may be viewed as a redundant task given that the content is already discussed with the patient.

There has been a surge in research on automatic creation of medical documentation from patient-doctor conversations. A lot of it is focused on extracting medical information and facts from the patient-doctor conversation (Happe et al., 2003; Quiroz et al., 2019). This could involve extracting clinical standard codes (Leroy et al., 2018), clinical entities such as symptoms, medications, and their properties (Du et al., 2019a,b), or medical regimen (Selvaraj and Konam, 2020). This extracted information could then be used to generate a report (Finley et al., 2018a,b). Such information extraction systems require creating an annotated conversation corpus (Patel et al., 2018a; Shafran et al., 2020).

For example, the NLP pipeline described by Finley et al. (2018a) first extracts knowledge from an

Proceedings of the 1st Workshop on NLP for Medical Conversations, pages 22–30
Online, 10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

ASR transcript and then generates the report. The knowledge extraction consists of tagging speaker turns and sentences with certain classes using RNN-based models, and using techniques such as string matching, regular expressions, and data-driven supervised and unsupervised approaches to extract information from the tagged sentences. This is followed by data-driven templates and finite state grammars for report generation.

We take a different approach where the problem is cast as translation (source language is conversational in nature and target language is clinical) and summarization (input contains redundant and irrelevant information, and target is a concise and precise note) at the same time. Given recent advances in neural transduction technology (Bahdanau et al., 2015; See et al., 2017; Vaswani et al., 2017), we explore the end-to-end paradigm for generating medical reports from ASR transcripts. This eliminates the need for annotated corpora that are required for training intermediate processing steps. As a result this approach is scalable across various medical specialties.

Sequence-to-sequence models have been used for summarizing radiology notes into the short *Impressions* section, possibly incorporating also other domain-specific information (Zhang et al., 2018; MacAvaney et al., 2019). In contrast, our system creates a report directly from the conversation transcript. Disadvantages of the end-to-end approach include that it limits the ability to inject prior knowledge and audit system output, and may potentially result in inferior performance.

2 Dataset

We use data consisting of ambulatory orthopedic surgery encounters. Speaker-diarized conversation transcripts corresponding to the audio files were obtained using an automatic speech recognizer. The reports for orthopedic surgery are organized under four sections—history of present illness (HPI), physical examination (PE), assessment and plan (AP), and diagnostic imaging results (RES). The HPI section captures the reason for visit, and the relevant clinical and social history. The PE section captures both normal and abnormal findings from a physical examination. The RES section outlines impressions from diagnostics images such as X-ray and CT scans. Finally, the AP section captures the assessment by the doctor and treatment plan e.g. medications, physical therapy etc.

	Size	**Source**		**Target**	
		Avg	**Max**	**Avg**	**Max**
Ortho HPI	802k	961	7,008	116	2,920
Ortho RES	444k	993	6,873	48	878
Ortho PE	769k	970	7,008	128	1,456
Ortho AP	811k	967	7,008	160	2,639
CNN&DM	287k	681	2,496	48	1,248
XSum	204k	431	33,161	23	432

Table 1: Statistics of our orthopedic report creation task and two other summarization tasks. Number of training examples and average and maximum number of tokens in the source and target sequence.

Experimental results are reported on a dataset that consists of around 800k encounters from 280 doctors. The dataset is partitioned chronologically (date of collection) into train, validation and evaluation partitions. The evaluation partition includes 4,000 encounters from 80 doctors. The doctors present in the evaluation set are present in the train set. Since the models do not require supervision outside the workflow, this paradigm is scalable, though future work will assess generalization to unseen doctors. We only use non-empty examples for training and evaluation. The RES section is empty in about 50 % of the examples.

Table 1 shows more detailed statistics of our dataset in terms of the number of training examples and source and target sequence lengths. The table also shows corresponding statistics for two prominent datasets for abstractive summarization that were not used in this study: CNN and Daily Mail, as processed by Nallapati et al. (2016), and XSum (Narayan et al., 2018). As shown in the table both the source and target sequences in our data are significantly longer than in the standard databases.

3 Modeling

We use neural sequence-to-sequence models for summarizing the doctor-patient conversations. The input to the model is a sequence of tokens generated by the speech recognizer. The medical reports consist of four sections, and we produce each section using a separate sequence-to-sequence model.

The task closely resembles machine translation, so we use models that are similar to neural machine translation models. There are, however, several differences from a typical machine translation task:

1. The source and target sequences are in the same language, thus we can use the same vo-

cabulary for input and output.

2. Report generation may require reasoning over a long span of sentences. The sequences, especially the source sequences, are significantly longer, since we cannot translate sentences separately.

3. Information may be incomplete (patient gesturing where it hurts), redundant (patient or doctor repeating information), or irrelevant conversation. In translation the semantic content in both the source and the target sequence is the same.

The models that we use are based on the encoder-decoder architecture that is well known from neural machine translation (Sutskever et al., 2014). All the models rely on attention (Bahdanau et al., 2015). The encoder creates context-dependent representations of the input tokens, and the decoder produces the next-token probability from those representations. During inference the output is generated autoregressively using the next-token probabilities.

Very long sequences increase the memory usage and training time, and make it more difficult to learn the model parameters. We truncate the source sequences to 2,000 tokens and the target sequences to 500 tokens during training. 10 % of the source sequences and 0.1 % of the target sequences were above this threshold. During inference we truncate the inputs to 3,000 tokens. Only 4 % of the test examples were originally longer than this limit.

In this work we compare models that are based on recurrent neural networks and models based on Transformer (Vaswani et al., 2017). The following sections describe these models and the enhancements that we have implemented.

3.1 RNN with Attention

The RNN sequence-to-sequence model with attention was introduced by Bahdanau et al. (2015). The encoder creates context-dependent input representations using a bidirectional RNN. The decoder produces the next-token probability using a unidirectional RNN, since future information is not available. We used LSTM (Hochreiter and Schmidhuber, 1997) as the recurrency mechanism.

We included in the model some of the enhancements from the RNMT+ model (Chen et al., 2018)—dropout, residual connections, layer normalization, and label smoothing. We also increased the number of encoder and decoder layers to two, but further increasing the number of layers did not give any benefit. We did not see significant benefit from using multi-head attention.

3.2 Hierarchical Encoder

Training the RNN model is slow due to their inherently sequential form precluding parallelization within the long input and output sequences. With longer input sequences it also becomes increasingly difficult for the model to learn to attend to relevant parts of the input.

Inspired by Cohan et al. (2018), we split the input sequence into 8 equal-length segments that are encoded independently. The segments can be processed in parallel, speeding up training considerably. The final LSTM hidden state from forward and backward directions of each segment are concatenated and projected into a segment embedding. After a stack of segment encoders, one more bidirectional LSTM runs over the segment embeddings of the previous layer (see Figure 1). The attention distribution is computed using both the token-level (second layer) and segment-level (third layer) outputs similar to Cohan et al. (2018)—the token-level scores are weighted by the normalized segment-level scores.

The hierarchical encoder sped up training by a factor of three with little to no impact on the summarization accuracy.

3.3 Pointer-Generator

To facilitate effective copying of parts of conversations to the output we implemented the pointer-generator network from See et al. (2017). It reuses the encoder-decoder attention distribution as a pointer for the copy mechanism. The same attention distribution is still used for computing the context vector and a probability distribution over the output vocabulary. The attention distribution is taken as a probability distribution over the input tokens and interpolated with the vocabulary distribution. The interpolation coefficient is learnt from the context vector, decoder LSTM state, and decoder input embedding. This mechanism also enables handling of words that are not present in the decoder vocabulary.

The pointer-generator network is illustrated in Figure 1. The context vector from attention is fed into a linear layer with the decoder state to produce the vocabulary distribution. The attention distribution is interpolated with the vocabulary distribution,

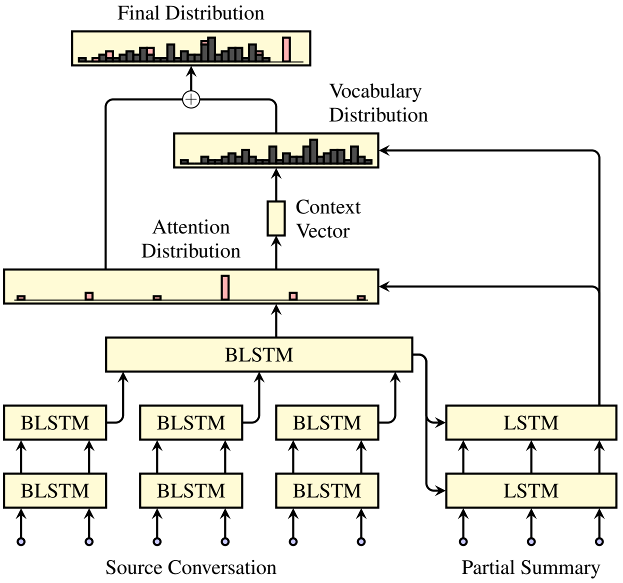

Figure 1: Illustration of the RNN sequence-to-sequence model with hierarchical encoder and pointer-generator copy mechanism. Segments of source conversation are encoded independently using two bidirectional LSTM layers, and a third layer runs over the final segment embeddings. The attention distribution is computed using both the token-level and segment-level outputs. The final distribution is interpolated from the attention distribution and the vocabulary distribution using a predicted coefficient.

although the connections for predicting the interpolation coefficient have been omitted from the figure for clarity.

The authors also introduce a coverage loss for training. They define coverage as the sum of attention weights over previous decoding steps. The coverage loss penalizes for attending to positions where the coverage is already high. The purpose is to encourage the model to attend to all input positions while decoding a sequence, and reduce repetition. We found the coverage loss to be somewhat helpful with a small weight (0.001).

3.4 Transformer

Transformer uses self-attention (Vaswani et al., 2017) in the encoder and decoder to create context-dependent representations of the inputs. In our experiments both encoder and decoder consist of six layers of self-attention. Each decoder layer attends to the top of the encoder stack after the self-attention. Additionally each encoder and decoder layer contains a position-wise feed-forward or convolutional network that consists of two transformations and a ReLU activation in between. The fact that these layers can be computed in parallel for every position makes training more efficient than training RNN models.

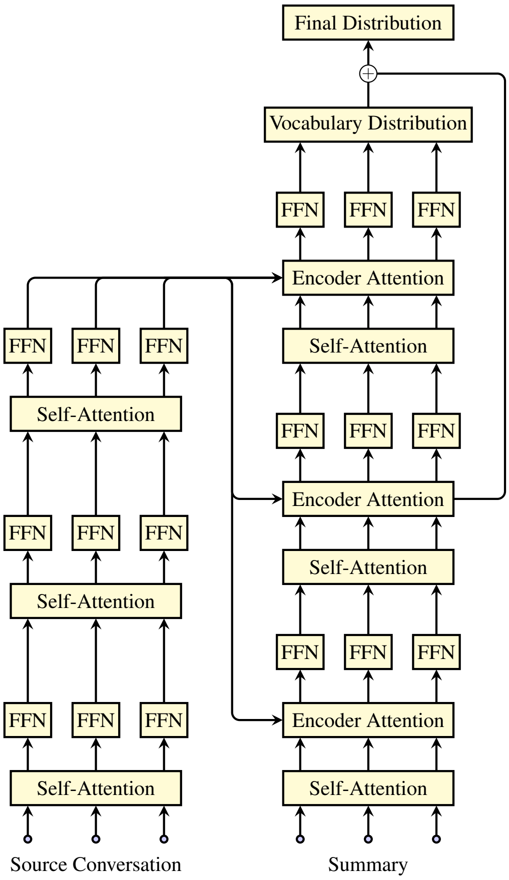

Figure 2: Illustration of the Transformer sequence-to-sequence model with pointer-generator copy mechanism. Each encoder layer consists of self-attention and a position-wise feed-forward network. Decoder layers also attend to the top of the encoder stack. We take one attention distribution from the penultimate decoder layer and interpolate it with the vocabulary distribution using a predicted coefficient. The final distribution includes the vocabulary tokens and the present source tokens. Layer normalization and residual connections are omitted for clarity.

Following Vaswani et al. (2017), we use the base model size, i.e. 8 attention heads with a total of 512 outputs and a 2048-dimensional feed-forward network. Following Domhan (2018), we apply layer normalization (Ba et al., 2016) before the self-attention and feed-forward sub-layers. This greatly stabilizes training and speeds up convergence with long inputs, confirming observations earlier made with deep networks (Wang et al., 2019).

Since the output of attention is independent of the order of the inputs, we inject position-dependent information into the inputs. In the origi-

	ROUGE-L RERR				Fact F_1 RERR			
	HPI	RES	PE	AP	HPI	RES	PE	AP
RNN	4.5	38.8	50.1	18.7	27.5	44.6	64.9	24.2
Hierarchical RNN	9.2	43.3	56.3	21.4	29.7	49.7	68.4	26.8
Hierarchical RNN + PG	9.2	45.4	53.7	22.3	30.8	51.3	67.1	29.2
Transformer	18.6	49.6	**65.4**	40.2	39.1	55.0	**74.6**	46.8
Transformer + PG	**19.2**	**51.0**	**65.4**	**42.0**	**39.5**	**56.7**	74.2	**49.4**

Table 2: Relative error rate reductions calculated from ROUGE-L and fact extractor F_1 scores. The scores are relative to an oracle baseline model that produces the longest common subsequence between the input and the reference output. The models labeled with PG use the pointer mechanism and coverage training loss.

nal paper, Vaswani et al. (2017) added sinusoidal position information before the first layer. We use relative position representations (Shaw et al., 2018) that are added inside the attention mechanism, which we found to work slightly better.

We also implemented a pointing mechanism in the Transformer model, similar to the RNN pointer-generator. For pointing we can use any distribution over the source tokens. The Transformer model creates several encoder-decoder attention distributions, one for each attention head in each layer. In principle any single head or the average of heads could be used for pointing. We argue that dedicating a single attention head should be sufficient, since the parameters of that head will be trained to attend to the tokens that are good candidates for copying. In this case the rest of the attention heads will not be affected and will perform their usual function, unlike when averaging over the attention heads. The penultimate layer seems to naturally learn alignments (Garg et al., 2019), so we use its first attention head for pointing. A simplified picture of the model is in Figure 2.

	HPI	RES	PE	AP
RNN	168	168	168	168
Hierarchical RNN	168	117	168	168
Hierarchical RNN + PG	168	131	168	168
Transformer	68	26	68	64
Transformer + PG	69	26	70	66

Table 3: Training time in hours on the four report sections for the various model architectures. Training time was restricted to one week, causing most RNN jobs to stop before reaching the maximum number of training steps.

4 Experiments

We train the models on Azure cloud using NVIDIA V100 GPUs. Each training job is distributed to 8 GPUs. We use data-parallel training, i.e. each GPU processes their share of the mini-batch and then the gradients are averaged over the GPU devices. The batch size is set to a maximum of 7,000 source tokens per GPU. NVIDIA NCCL library is used to perform the communication efficiently.

We use a vocabulary consisting of the 10k most frequent words. The same vocabulary is shared between the source and target tokens.

We use Nesterov's Accelerated Gradient (Nesterov, 1983) with the RNN models, while Adam (Kingma and Ba, 2015) is found to perform better with the Transformer models. We train the models a maximum of 400k steps, excepting RES section models, which are trained until 200k steps due to their fewer examples and shorter targets. This corresponds to approximately 25 epochs on RES section and 30 epochs on other sections. During this time we observe that training has practically converged and training longer would not provide significant benefit. We also limit individual model training to one week as a cost control.

Improved performance is obtained via averaging model parameter from 8 checkpoints, with interval length as a function of total training steps. Where helpful, we use a cyclical learning rate schedule, with the cycle length set to the checkpoint saving interval, so that the saved checkpoints would correspond to the minimums of the learning rate schedule (Izmailov et al., 2018).

4.1 Results

ROUGE (Lin, 2004) is a collection of metrics designed for evaluation of summaries. We calculate ROUGE-L, which is an F_1 score that is based on the lengths of the longest common subsequences

Partial ASR Transcript

[doctor]: we'll do a celebrex refill let me see you back four to six months earlier if needed okay hey good to see you good to see you
[patient]: thank you thank you thank you

Reference Output

i have refilled her celebrex to have available. the patient will follow up in four to six months or earlier if needed.

Baseline Model Output

celebrex four to six months earlier if needed

Transformer PG Model Output

i have provided the patient with a refill of celebrex. the patient will follow up in 4-6 months or sooner if needed.

Figure 3: An excerpt of a speaker-diarized ASR transcript, its reference AP section output, the baseline model output, and partial output of the Transformer pointer-generator model (all lower-cased, without formatting). The baseline model produces the longest common subsequence between the transcript and the reference output.

	HPI	RES	PE	AP
RNN	12.1	0.4	3.8	19.9
Hierarchical RNN	2.0	0.2	1.2	15.8
Hierarchical RNN + PG	3.6	0.1	1.3	16.9
Transformer	2.0	0.1	1.2	0.3
Transformer + PG	1.7	1.0	0.8	0.3

Table 4: Percentage of model output considered part of a repetition. We define repetition as a sentence that occurs at least four times in the same report, or an n-gram of at least 16 tokens that repeats consecutively.

between the reference and hypothesis sentences. We have noticed that it measures the fluency of the language well. However, we are also interested in assessing factual correctness. For this we utilize a proprietary machine-learning-based clinical fact extractor. It is capable of extracting medical facts such as conditions and medications, as well as their attributes such as body part, severity, or dosage. We extract facts from the model output and the ground-truth report, and compute the F_1 score from these two sets.

We publish our scores relative to an oracle baseline model, which extracts the longest common subsequence between the input conversation and the reference output. An example of such output is in Figure 3. Table 2 shows the relative error rate reduction (RERR) from ROUGE-L and fact extractor F_1 scores. We define the error rate as the complement $(1 - s)$ of the original score.

The Transformer models clearly obtain better scores than any of the RNN models. Partly this is because most RNN experiments are limited by the maximum training time. In the RES section both hierarchical RNN and Transformer models reached 200k training steps, but Transformer performance is still superior. An example output generated by the Transformer pointer-generator model is shown in Figure 3.

The training times are shown in Table 3. All but one of the RNN experiments were stopped after reaching the one week limit. Normal RNN training was terminated after approximately 50k steps, while hierarchical RNN progressed 160k–230k steps over the same duration. Performance was similar at an equal number of steps, but given fixed practical time and cost constraints, the hierarchical encoder yields improved results.

The pointer mechanism generally provided a small performance boost, with the largest improvements in RES and AP section quality. Interestingly, the pointer mechanism can even hurt performance in PE section. This is exaggerated with the RNN models by the fact that the pointer-generator model is slower to train and progressed only 175k steps, while the same model without the pointer mechanism and coverage loss progressed 225k steps in the time limit. Generally the ROUGE-L and fact F_1 scores seem correlated, displaying similar differences across models.

4.2 Repetition

By visual inspection of generated reports we noticed that some models suffer from an excessive amount of repetitions. We identified two main categories: sentences that occur multiple times in the same report and consecutively repeating n-grams. We try to assess the amount of repetition in model output by detecting these two types of patterns. Not all occurrences of such patterns are mistakes, however, and even the reference targets contain such patterns. We limit to sentences that occur at least 4 times and repeating n-grams that are at least 16 tokens long. Table 4 shows the repetition rates in model outputs as the percentage of tokens that fall into either of these categories. The table shows that the problem diminishes when training longer.

In the reference reports there are only a few instances of tokens that we consider repetitive, and these appear to be mistakes by the writer of the report. We should then aim at 0 % repetition rate. Note that the purpose of this metric is not to detect language where for example the same frequent words are used more often than in natural language. We rather wanted to assess how widely the models suffer from artificial and clearly erroneous repetition of word sequences.

5 Conclusions

In this paper we compared RNN and Transformer-based sequence-to-sequence architectures for medical report generation from patient-doctor conversations. This study demonstrates the ability of sequence-to-sequence models, in particular Transformer, to not only extract relevant clinical conversation excerpts, but abstractively summarize in a relatively fluent and factually correct medical report. Especially when working within compute and time budgets, Transformer is superior to traditional RNN-based models, and scalable to large datasets.

Visual inspection showed that commonly occurring problems in the generated reports included repeated sentences and hallucinated clinically consistent sentences unfounded by the conversations. Minimally a human would need to be in the loop to verify or correct these machine-generated reports. Future work includes comparing end-to-end approaches with a pipeline of clinical information extraction and natural language generation methods.

References

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. In *NIPS 2016 Deep Learning Symposium*, Barcelona, Spain.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George F. Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 76–86. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Tobias Domhan. 2018. How much attention do you need? A granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808, Melbourne, Australia. Association for Computational Linguistics.

Nan Du, Kai Chen, Anjuli Kannan, Linh Tran, Yuhui Chen, and Izhak Shafran. 2019a. Extracting symptoms and their status from clinical conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 915–925, Florence, Italy. Association for Computational Linguistics.

Nan Du, Mingqiu Wang, Linh Tran, Gang Lee, and Izhak Shafran. 2019b. Learning to infer entities, properties and their relations from clinical conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4979–4990, Hong Kong, China. Association for Computational Linguistics.

Gregory Finley, Erik Edwards, Amanda Robinson, Michael Brenndoerfer, Najmeh Sadoughi, James Fone, Nico Axtmann, Mark Miller, and David Suendermann-Oeft. 2018a. An automated medical scribe for documenting clinical encounters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–15, New Orleans, Louisiana. Association for Computational Linguistics.

Gregory Finley, Erik Edwards, Amanda Robinson, Najmeh Sadoughi, James Fone, Mark Miller, David Suendermann-Oeft, Michael Brenndoerfer, and Nico Axtmann. 2018b. An automated assistant for medical scribes. In *Proceedings of Interspeech 2018*, pages 3212–3213.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with Transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Andr Happe, Bruno Pouliquen, Anita Burgun, Marc Cuggia, and Pierre [Le Beux]. 2003. Automatic concept extraction from spoken medical reports. *International Journal of Medical Informatics*, 70(2):255 – 263. MIE 2002 Special Issue.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):17351780.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885. AUAI Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Gondy Leroy, Yang Gu, Sydney Pettygrove, Maureen K Galindo, Ananyaa Arora, and Margaret Kurzius-Spencer. 2018. Automated extraction of diagnostic criteria from electronic health records for autism spectrum disorders: Development, evaluation, and application. *Journal of Medical Internet Research*, 20(11):e10497.

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.

Sean MacAvaney, Sajad Sotudeh, Arman Cohan, Nazli Goharian, Ish Talati, and Ross W. Filice. 2019. Ontology-aware clinical abstractive summarization. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR19, pages 1013–1016, New York, NY, USA. Association for Computing Machinery.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376.

Maria Panagioti, Efharis Panagopoulou, Peter Bower, George Lewith, Evangelos Kontopantelis, Carolyn Chew-Graham, Shoba Dawson, Harm van Marwijk, Keith Geraghty, and Aneez Esmail. 2017. Controlled interventions to reduce burnout in physicians: A systematic review and meta-analysis. *JAMA Internal Medicine*, 177(2):195–205.

Pinal Patel, Disha Davey, Vishal Panchal, and Parth Pathak. 2018a. Annotation of a large clinical entity corpus. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2033–2042, Brussels, Belgium. Association for Computational Linguistics.

Rikinkumar S Patel, Ramya Bachu, Archana Adikey, Meryem Malik, and Mansi Shah. 2018b. Factors related to physician burnout and its consequences: A review. *Behavioral sciences*, 8(11):98.

Thomas H. Payne, W. David Alonso, J. Andrew Markiel, Kevin Lybarger, and Andrew A. White. 2018. Using voice to create hospital progress notes: Description of a mobile application and supporting system integrated with a commercial electronic health record. *Journal of Biomedical Informatics*, 77:91–96.

Juan C. Quiroz, Liliana Laranjo, Ahmet Baki Kocaballi, Shlomo Berkovsky, Dana Rezazadegan, and Enrico Coiera. 2019. Challenges of developing a digital scribe to reduce clinical documentation burden. *npj Digital Medicine*, 2(1):114.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Sai Prabhakar Pandi Selvaraj and Sandeep Konam. 2020. Medication regimen extraction from clinical conversations. In *International Workshop on Health Intelligence (W3PHIAI 2020)*, New York, USA.

Izhak Shafran, Nan Du, Linh Tran, Amanda Perry, Lauren Keyes, Mark Knichel, Ashley Domin, Lei Huang, Yu hui Chen, Gang Li, Mingqiu Wang, Laurent El Shafey, Hagen Soltau, and Justin Stuart Paul. 2020. The medical scribe: Corpus development and model performance analyses. In *Proceedings of the Twelfth International Conference on Language*

Resources and Evaluation (LREC 2020), Marseille, France. To appear.

Tait D. Shanafelt, Liselotte (Lotte) Dyrbye, Christine Sinsky, Omar Hasan, Daniel Satele, Jeff A. Sloan, and Colin Patrick West. 2016. Relationship between clerical burden and characteristics of the electronic environment with physician burnout and professional satisfaction. *Mayo Clinic Proceedings*, 91(7):836–848.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep Transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Alexi A. Wright and Ingrid T Katz. 2018. Beyond burnout—redesigning care to restore meaning and sanity for physicians. *The New England Journal of Medicine*, 378(4):309–311.

Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D. Manning, and Curtis P. Langlotz. 2018. Learning to summarize radiology findings. In *EMNLP 2018 Workshop on Health Text Mining and Information Analysis*.