Interspeech 2022
18-22 September 2022, Incheon, Korea

Extract and Abstract with BART for Clinical Notes from Doctor-Patient Conversations

Jing Su, Longxiang Zhang, Hamid Reza Hassanzadeh, Thomas Schaaf

3M | M∗Modal

{jsu6, lzhang28, hhassanzadeh, tschaaf}@mmm.com

Abstract

Reducing the burden of documentation physicians are required to do with speech understanding is a challenging and worthwhile goal with the potential to improve care. When transcripts of doctor-patient conversations are available, automatic summarization with deep neural networks is one promising solution to reducing documentation workload. We develop an "extract-and-abstract" approach to automatic generation of the History of Present Illness (HPI) section in clinical notes with BART: we train a classifier on annotated data to predict a clinical section each utterance is most relevant to; we then utilize the trained classifier to select only utterances from conversations relevant to HPI to be considered as input to BART for summarization; we experiment with additional filtering methods on selected utterances to further reduce input truncation due to the token limit of BART model. Results show that the generated summaries from our approach improve in both ROUGE scores and extracted medical concepts over previously published results. Considering the improvement is achieved with a relatively small set of doctor-patient conversations, we expect further improvement with more labeled data in the future.

Index Terms: medical conversations, topic detection, conversation summarization, natural language generation

1. Introduction

In recent years, automatic abstractive summarization of **Do**ctor-**Pa**tient **Co**nversations (DoPaCos) into clinical notes has gained momentum in research in both healthcare and machine learning communities [1, 2, 3, 4]. It builds on the recent success of deep neural network based models (PGNet[5], T5[6], Pegasus[7], BART[8], to name just a few) on the summarization task of public domain data such as news articles, and offers a potent solution to reducing documentation workload of both physicians and medical scribes working with modern Electronic Health Records (EHR) systems [9].

Conversations such as DoPaCos exhibit natural topic-segmented structure: for documentation purpose, doctors tend to guide the conversation topic in order to obtain required information in a loosely stable order from the patient (e.g., diagnosis followed by assessment then plan). There is extensive study on incorporating topic structures in the training of summarization models: Topic-aware PGNet [10] assigned topic-level attention weights on the pointer side of PGNet during training; Dr Summarize [3] produced a topic specific snippet dataset from complete conversation to help guide the training of PGNet model; QMSum [11] forced model training to generate relevant summaries from meeting notes based on a topic query. For DoPaCo summarization in particular, the section structure of EHR and clinical notes has also been a source of prior knowledge people inject into the model training process: Krishna et al [4] trained a classifier to first extract "noteworthy" utterances from input conversation for each note section, clustered them and then adopted a summarizer to generate single summary sentence from each cluster when generating a SOAP note [12]. Zhang et al [13] focused on History of Present Illness (HPI) section summarization and divided input conversation into chunks in a two-stage framework to better utilize long context as well as to overcome the input length limit of transformer-based models.

In this paper, we propose an *extract-and-abstract* approach to the task of summarizing HPI section of a clinical note from DoPaCos, inspired by [4, 13]. Unlike the automatic algorithm in [13] or the requirement of extensive human annotations with a large label set in [4], our method incorporates a lightweight utterance labeling task with only four clinical section labels; a classifier is trained on the annotated corpus and is employed in the selection of HPI-relevant utterances (*extract*); only the selected utterances are considered as candidate input for training a summarization model (*abstract*). Different from [4], we trained the summarization model to generate complete HPI notes instead of individual sentences. To alleviate input truncation limited by the transformer model (1024 tokens by BART), we propose both *Adaptive Thresholding* and *Slicing* methods to filter additional utterances selected by the extractive classifier while maintaining the most relevant information. We hypothesize that such an *extract-and-abstract* approach should help summarization by removing distracting information from DoPaCos, along with an easy-to-implement pipeline design in mind (Figure 1).

2. Dataset

We use two disjoint datasets in this paper to accommodate the *extract* and *abstract* components in our proposed approach. For training our utterance selection models (Section 4.1), We employed in-house linguists with medical knowledge to label each utterance in DoPaCos with one of three *Span Labels*: *iphi, pe, a/p*, which stand for inclusive HPI[1], Physical Examination and Assessment & Plan, respectively. See Table 1 for an example. We coined the term *Span Label* to emphasize the requirement for our annotators that each label should span at least three consecutive utterances in one conversation[2]. This reflects the intrinsic continuity of topical information within neighboring utterances, while keeping utterances of one annotated class linguistically coherent to a certain degree. The final annotated corpus contains 915 de-identified DoPaCo transcripts, which we split by 606(292)-161(240)-148(262) into *train-dev-test* sets. The numbers in parentheses mark the average number of utterances per conversation in each set. In training set the numbers of utter-

[1] Contents in conversations that we deem relevant to history of present illness and review of system of the patients.

[2] We also defined a separate set of labels for more localized information, usually within one utterance. The complete annotation framework, guideline design, and dataset quality analysis is detailed in a separate paper in the process of publication.

10.21437/Interspeech.2022-10935

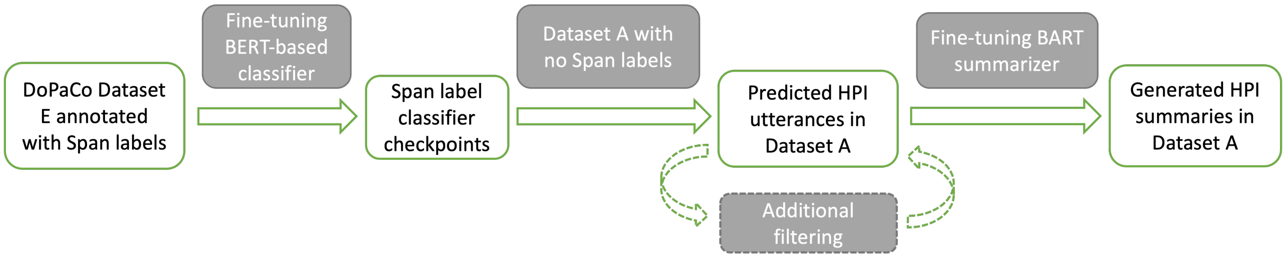
DoPaCo Dataset E annotated with Span labels
Fine-tuning BERT-based classifier
Span label classifier checkpoints
Dataset A with no Span labels
Predicted HPI utterances in Dataset A
Additional filtering
Fine-tuning BART summarizer
Generated HPI summaries in Dataset A

Figure 1: *Pipeline of supervised learning approach in utterance selection and summarization.*

ances for each class are: 68426 (*ihpi*), 9809 (*pe*), 76412 (*a/p*), and the remaining utterances are given a null label. We refer to this corpus as Dataset E.

Dataset A, which we used for fine-tuning our summarization model in the *abstract* component (Section 3.2), contains a total of 1342 de-identified DoPaCos with an average of 16 human written HPI summaries per conversations, which we split into 939(15043)-201(3095)-202(3450) for *train-dev-test*. The numbers in parentheses are the number of reference summaries in each set. All summaries were created by medical scribes listening to the conversation audios and recorded via an in-house simulated EHR system. We choose to focus on HPI section for summarization due to reported success by [13] and the fact that scribes are instructed in this section to write coherent paragraphs which are suitable summarization targets. Dataset A has no overlapping conversations with Dataset E. We believe this setup should prevent the utterance selection models overfitting to the summarization data, and can test the flexibility of data requirement in our proposed *extract-and-abstract* approach.

3. Methods

We propose an *extract-and-abstract* pipeline (Figure 1) to the task of HPI summarization of DoPaCos. Two models: utterance selection model and summarization model (the first and third steps in Figure 1) are of key importance in our approach.

3.1. Utterance selection models

The goal of the utterance selection model is to filter utterances unrelated to HPI from DoPaCos by predicting the span label of each utterance. Such a task can be solved as either (a) sentence classification or (b) sequence tagging. We experimented with both paradigms, and described as follows the details of three utterance selection models trained on Dataset E.

3.1.1. Sentence classifier

The baseline model we experimented with is a BERT-based [14] neural network with MLP layers and a 4-way softmax layer on top (BERT+MLP); the input to the MLP layers is the vector representation of the [CLS] token from the top layer of BERT. We treated the utterance selection as simple sentence classification problem and fine-tuned the model (both BERT and MLP layers) on individual utterances prefixed by speaker roles, e.g. *[DR]: how are you today?*, with the associated span label as targets. This is also referred to later on as a *Context (N=0) model.*

3.1.2. Context model

In order to incorporate contextual information which is key to utterance label prediction, while maintaining the task as sentence classification, we adopted the same BERT+MLP model architecture as the *Context (N=0) model*, but changed the input from a single utterance to concatenation of several consecutive utterances, and the target was chosen as the label of the last utterance. Furthermore, we concatenated all utterances from all conversations in the training set and used a sliding window scan with stride 1 to generate training examples. We set the maximum input length to be 128 tokens and tuned the window length of the context, i.e. the number of preceding utterances to include, within a limited set of values ($N <= 9$). Four leading utterances lead to the best performing classifier, which is reported in this paper and we refer to it as *Context (N=4) model* in the following text.

3.1.3. Sequence model

We experimented with training the utterance selection model as a sequence tagging problem, which is more aligned with the annotation task and takes more direct account of both conversation context and correlation between labels. The model architecture is BERT+LSTM+CRF with a sequence of utterances as input. Each utterance is fed through the BERT model to obtain the token-level vector representation, which is then averaged across all tokens into an utterance level representation. The sequence of utterance-level representations is then passed through the two LSTM [15] layers with hidden dimensions tuned to 50 followed by a linear layer to project the representation of each utterance down to a 4-dimensional vector. Finally, the Linear-chain CRF layer [16, 17] receives the vector representations and predicts the label of all utterances in the sequence. The target is the corresponding sequence of ground truth span labels. During training, we froze the weights of the BERT model and only fine-tuned weights in the rest of the model.[3] The length of the sequence (number of utterances) was chosen to be 30 by hyperparameter tuning; samples were generated using the same sliding-window approach as with *Context (N=4) model* with window size 30 and stride 1.

Since the trained model predicts a label for every utterance in the input sequence, at inference time one utterance will get multiple predictions from all sequences that contain it. The final label for the utterance is determined by majority voting. We denote this utterance selection model as *Sequence model.* During training, we observed that using model weights of pre-trained BERT led to weak model performance. This observation is not surprising considering off-the-shelf BERT is pre-trained on public domain text such as news and there is a significant domain shift to medical conversations. We therefore chose to

[3] End-to-end finetuning of BERT+LSTM+CRF would require several passes of the BERT model for each input sequence, and can easily lead to out-of-memory error in a straightforward implementation.

use the model weights from BERT fine-tuned in our best performing context model as described in Section 3.1.2 and the final results in Table 2 showed this yields the best performing utterance selection model.

Table 1: *An example conversation with annotated span labels*

Conversation Transcript	Span labels
...	
[DR]: lot of pain little pain no pain?	ihpi
[PT]: no pain.	ihpi
...	
[DR]: so there's a couple ways we can play this.	a/p
[DR]: I think you're right on track with it.	a/p
[DR]: I'd just keep that up.	a/p
...	

3.2. Summarization model

We adopted BART [8] as our summarization model based on its previous success in summarizing DoPaCos [13]. We settled with the BART-large model architecture and conducted fine-tuning of BART model on Dataset A in the same way as the single-stage fine-tuning as detailed in [13]. One major difference from [13] is that we fine-tuned the model using all available reference summaries in Dataset A instead of using a single reference per conversation.

One important consideration (third module in Figure 1) in our proposed *extract-and-abstract* approach is the filtering of utterances in DoPaCo based on span labels predicted by the utterance selection model. Intuitively, one would consider keeping all utterances predicted by the model as *ihpi* and concatenating them as the input to BART. However, we observed that 31.4% of the conversations exceed the 1024 token limit for BART model, even after filtering non-*ihpi* utterances (Figure 2 (a)). Therefore, we applied the following additional filtering methods to circumvent the truncation problem[4]:

Direct Truncation. For each C_t, we keep selecting each utterance in order until the combined word count exceeds the token limit. The remaining utterances are discarded.

Adaptive Thresholding (*Ada. Thr. N*). For each C_t and seven different threshold values $T = [0.7, 0.8, 0.9, 0.95, 0.96, 0.97, 0.98]$, we keep only utterances with $P(S) > T$; the seven threshold values are applied sequentially until either the remaining combined utterances are within the preset input limit ($N = 320$ or $N = 640$ words) or all seven threshold values have been exhausted. This adaptive thresholding is only applied to filtered conversations with $|C_t| > 1024$.

Slicing. For filtered conversations with $|C_t| > 1024$, we split the utterances into consecutive chunks $C_{t_1}, C_{t_2}, \cdots, C_{t_k}$ such that $|C_{t_i}| < 1024$. We add special token '—' at the beginning of $C_{t_2}, \cdots, C_{t_k}$ to indicate this is not a new conversation. Each chunk is matched to the same reference summary and used as separate samples in training BART. During inference, only C_{t_1} is adopted and we get one summary for each DoPaCo.

Figure 2 shows histogram of word count of the filtered conversations with (Figure 2(b) - 2(c)) additional filtering and without (Figure 2 (a)). We can see the truncation problem persists with filtered conversations, but is basically avoided with additional filtering. For comparison, we also considered two model-free utterance selection strategies: using the first 320/640 words ($\approx$ 512/1024 tokens) in a conversation as the input to BART (**Single stage first 320/640**); and using the last 640 words (**Single stage last 640**)).

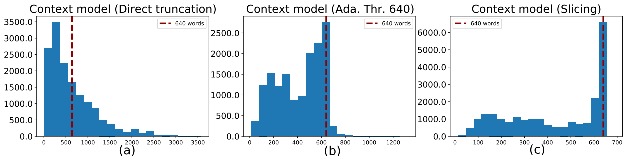

Figure 2: *Word count histogram per training example (conversation) for the summarizer. Complete utterances are selected by (a) ihpi classifier predictions. (b) Adaptive Thresholding. (c) Slicing. The red vertical line shows the 640 word limit of summarizer input.*

4. Experiments & Results

In this section we discuss the performance and evaluation of the utterance selection model and BART. For utterance selection models, we report F1 scores for all three span label classes; for summarization (BART), we choose ROUGE [18] to measure the similarity between generated and reference summaries. Since multiple references exist for every DoPaCo, we report a *mean-of-mean* rouge score similar to [13][5]. In addition, we choose to include a concept-based metrics (precision, recall, and F1) using concepts extracted from both reference and generated summary by **quickUMLS**[6], which is a Python implementation of Unified Medical Language Systems (UMLS)[7]. We believe this metrics complements the lexical similarity measured by ROUGE with a focus on the agreement in key medical information.

4.1. Utterance selection models

Table 2 lists the F1 scores for all span label classes across three utterance selection models proposed in Section 3.1. We can see that both *Context model (N=4)* and *Sequence model* improve greatly above the baseline *Context model (N=0)*, with 10% and 12% relative improvement in F1 scores for *ihpi* class, respectively; the F1 scores on the other two minor classes are also much improved, indicating improved model robustness against class imbalance. Since the downstream task in summarization was focused on HPI section, we were mostly interested in the model performance in *ihpi* class and decided to consider both *Context model (N=4)* and *Sequence model* as utterance selector in our summarization pipeline.

In Section 3.2 we introduce multiple additional filtering methods over utterances predicted with *ihpi* label. In the case of Adaptive Thresholding, we chose to apply it only to predictions made by *Context model (N=4)*. This is because *Sequence model* applied majority voting when predicting utterance labels, and calculating class probabilities of one utterance present in multiple sequences is not as straightforward as in the *Context model (N=4)*.

[4] To simplify notation, we denote probability thresholds as T, utterances as S, one conversation with *only predicted ihpi utterances* as C_t, $|C_t|$ as the token count of the filtered conversation, and $P(S)$ as the *ihpi*-class probability output by the utterance selection model on utterance S.

[5] Mean-of-mean rouge: average the rouge scores between generated summary and all reference summaries for one conversation, and then average again across all conversations.

[6] https://github.com/Georgetown-IR-Lab/QuickUMLS

[7] https://www.nlm.nih.gov/research/umls/index.html

Table 2: *Span label prediction F1 scores on Dataset E test set. (Note: w. avg. stands for weighted average.)*

	a/p	**pe**	**ihpi**	**w. avg.**
Context model (N=0)	0.74	0.49	0.64	0.67
Context model (N=4)	0.82	0.76	0.74	0.78
Sequence model	0.84	0.79	**0.76**	0.81

4.2. Evaluation

Table 3 records ROUGE scores of BART generated summaries given different utterances selected by the utterance selection model. quickUMLS evaluation results are presented in Table 4. There are four groups of experiments in both tables. The first group shows results from the Direct truncation and Slicing method. The second and third groups feature the Adaptive Thresholding method over 320 and 640 word limits with side-by-side results from model-free selection strategies with the same word limits. The last group is cited from [13] and reflects BART performance by using a single reference per DoPaCo during training. As can be seen, our models consistently improve over [13] by both ROUGE and quickUMLS evaluation.

In the first group, the Slicing method on *Context model (N=4)* shows the best ROUGE scores among all model-based filtering methods although its advantage is marginal (0.3% in ROUGE-L F1 over the runner-up). Adaptive Thresholding with the first 640 words on *Context model (N=4)* achieves better ROUGE scores than Direct truncation of both utterance selection models. The same trend also holds with quickUMLS metrics: Slicing on *Context model (N=4)* yields the best F1 with more than 2.2% increase over the second-best model (Sequence model with Direct truncation). The sequence model does not show an advantage over the context model although it performs the best in span label prediction.

Comparing groups two and three, we find that models based on the first 640 words of a conversation always perform better in both ROUGE and quickUMLS than their counterparts using only the first 320 words. In addition, the worst-performing model in both metrics is the *Single stage last 640* baseline. We believe this indicates that BART can benefit from a longer context in the input, but is also exposed to a position bias that tends to favor early utterances in the conversation. This agrees with our domain knowledge that most diagnosis of a patient's condition tends to occur early in DoPaCos.

One finding to our surprise is that the *Single stage first 640* baseline achieves the highest ROUGE-2 and ROUGE-L scores; this seems to contradict findings in other work (e.g. [4]) that filtering non-relevant information from input should help improve summarization. Although our utterance selection models may contribute to errors in the filtering of non-ihpi utterances, we believe the incoherence or conversational "gaps" in the filtered utterances is also a confounding factor that may diminish the effectiveness of the *extract-and-abstract* approach. However, this finding needs to be taken with a grain of salt, as we also observed that the baseline model doesn't perform as well as our Slicing method on *Context model (N=4)* in terms of concept-based evaluation (Table 4, 4th and 8th row), showing as much as 3.4% decrease in the precision score. One may argue that the Slicing method may benefit from an augmented training set since all sliced chunks were included in the training as separate samples, but we didn't observe similar improvement when applying the method to *Sequence model.* A more probable explanation could be different chunks provide different "views" of the same conversation during the BART training, and can guide the model to be more robust against the lexical variability of certain medical information (e.g. description of a symptom) in conversations.

Table 3: *"Mean-of-mean" ROUGE scores [13] over test set. * denotes the Baselines; R-1, R-2 and R-L stand for ROUGE-1, ROUGE-2 and ROUGE-L respectively.*

	R-1 F1	**R-2 F1**	**R-L F1**
Sequence model (Direct truncation)	0.3283	0.1173	0.3367
Sequence model (Slicing)	0.3276	0.1184	0.3351
Context model (N=4) (Direct truncation)	0.3349	0.1193	0.3417
Context model (N=4) (Slicing)	**0.3371**	0.1198	0.3445
Context model (N=4) (Ada. Thr. first 320)	0.3232	0.1126	0.3302
Single stage first 320 *	0.3259	0.1163	0.3356
Context model (N=4) (Ada. Thr. first 640)	0.3370	0.1202	0.3420
Single stage first 640 *	0.3362	**0.1246**	**0.3449**
Single stage last 640 *	0.3056	0.1040	0.3139
Multistage Chunking [13]	0.3227	0.1144	0.3302
Single stage [13]	0.3131	0.1097	0.3281

Table 4: *quickUMLS evaluation of BART-Large models over the Test set. (Note: * denotes the Baselines in this study. Ada. Thr. stands for Adaptive Thresholding.)*

	F1	**P**	**R**
Sequence model (Direct truncation)	0.4314	0.6452	0.3871
Sequence model (Slicing)	0.4201	0.6227	0.3742
Context model (N=4) (Direct truncation)	0.4303	0.6303	0.3850
Context model (N=4) (Slicing)	**0.4534**	**0.6596**	**0.4069**
Context model (N=4) (Ada. Thr. first 320)	0.4028	0.6287	0.3496
Single stage first 320*	0.4376	0.6317	0.4007
Context model (N=4) (Ada. Thr. first 640)	0.4224	0.6295	0.3718
Single stage first 640*	0.4440	0.6251	0.4049
Single stage last 640*	0.3615	0.5354	0.3202
Multistage Chunking [13]	0.4052	0.5316	0.3948
Single stage [13]	0.4093	0.5212	0.4009

5. Conclusions

We investigated an *extract-and-abstract* approach to automatic summarization of History of Present Illness from doctor-patient conversations. We utilized a corpus of DoPaCos with utterances annotated with clinical section labels to train an utterance selection model and employed the trained model in filtering irrelevant utterances in the input to the downstream summarization task with BART model. We proposed Adaptive Thresholding and Slicing methods as optional additional filtering steps and showed that using Slicing to further filter utterances relevant to HPI can lead to improved ROUGE and concept F1 scores.

Although we didn't observe a consistent improvement of our approach over the baseline approach of truncating input conversations, the *extract-and-abstract* approach did show improvement in concept-based evaluation with comparable ROUGE scores. This improvement indicates the generated summaries from our *extract-and-abstract* approach achieve better coverage of critical medical information. Given the limited data for our experiments, we believe our findings encourage further investigation of this approach in the domain of automatic medical summarization; and its effectiveness could be better supported with improved datasets and different summarization targets.

6. References

[1] N. Du, K. Chen, A. Kannan, L. Tran, Y. Chen, and I. Shafran, "Extracting symptoms and their status from clinical conversations," *arXiv preprint arXiv:1906.02239*, 2019.

[2] G. Finley, E. Edwards, A. Robinson, M. Brenndoerfer, N. Sadoughi, J. Fone, N. Axtmann, M. Miller, and D. Suendermann-Oeft, "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*, 2018, pp. 11–15.

[3] A. Joshi, N. Katariya, X. Amatriain, and A. Kannan, "Dr. summarize: Global summarization of medical dialogue by exploiting local structures," *CoRR*, vol. abs/2009.08666, 2020. [Online]. Available: https://arxiv.org/abs/2009.08666

[4] K. Krishna, S. Khosla, J. P. Bigham, and Z. C. Lipton, "Generating SOAP notes from doctor-patient conversations," *CoRR*, vol. abs/2005.01795, 2020. [Online]. Available: https://arxiv.org/abs/2005.01795

[5] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *CoRR*, vol. abs/1704.04368, 2017. [Online]. Available: http://arxiv.org/abs/1704.04368

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[7] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: pre-training with extracted gap-sentences for abstractive summarization," *CoRR*, vol. abs/1912.08777, 2019. [Online]. Available: http://arxiv.org/abs/1912.08777

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[9] R. S. Evans, "Electronic health records: then, now, and in the future," *Yearbook of medical informatics*, vol. 25, no. S 01, pp. S48–S61, 2016.

[10] Z. Liu, A. Ng, S. L. S. Guang, A. T. Aw, and N. F. Chen, "Topic-aware pointer-generator networks for summarizing spoken conversations," *CoRR*, vol. abs/1910.01335, 2019. [Online]. Available: http://arxiv.org/abs/1910.01335

[11] M. Zhong, D. Yin, T. Yu, A. Zaidi, M. Mutuma, R. Jha, A. H. Awadallah, A. Celikyilmaz, Y. Liu, X. Qiu, and D. R. Radev, "Qmsum: A new benchmark for query-based multi-domain meeting summarization," *CoRR*, vol. abs/2104.05938, 2021. [Online]. Available: https://arxiv.org/abs/2104.05938

[12] S. Zierler-Brown, T. R. Brown, D. Chen, and R. W. Blackburn, "Clinical documentation for patient care: models, concepts, and liability considerations for pharmacists," *American Journal of Health-System Pharmacy*, vol. 64, no. 17, pp. 1851–1858, 2007.

[13] L. Zhang, R. Negrinho, A. Ghosh, V. Jagannathan, H. R. Hassanzadeh, T. Schaaf, and M. R. Gormley, "Leveraging pretrained models for automatic summarization of doctor-patient conversations," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3693–3712. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.313

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

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[16] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[17] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Found. Trends Mach. Learn.*, vol. 4, no. 4, p. 267–373, apr 2012. [Online]. Available: https://doi.org/10.1561/2200000013

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