A Experimental Setup

A.1 Pre-trained Language Models

To test the generalizability of our method described in Section 3, we evaluate the efficacy of MEDVOC on three State-of-the-art encoder-decoder-based PLMs.

- **BertSumAbs** (BSA) [Liu and Lapata, 2019]: It uses a standard encoder-decoder framework where Bert acts as an encoder, and the decoder is six-layered transformer architecture initialized randomly[2]. BertSumAbs has 180 Million parameters, uses the *Word-Piece* tokenizer and its pretraining objective is a combination of *Masked Language Modeling* and *Next Sentence Prediction*. The vocabulary size of this PLM ($|V_{\text{PLM}}|$) is 30522.
- **BART** [Lewis *et al.*, 2020]: BART is a denoising autoencoder, implemented as a sequence-to-sequence model with a bidirectional encoder over corrupted text and a left-to-right auto-regressive decoder to generate the original document it was derived from. We use the BART-LARGE[3] model available from the *huggingface* library. BART has 406 Million parameters, uses *Byte-Pair Encoding* tokenization, and its pretraining objective is a combination of *Text Infilling* and *Sentence Shuffling*. The vocabulary size of this PLM ($|V_{\text{PLM}}|$) is 50265.
- **PEGASUS** [Zhang *et al.*, 2020a]: PEGASUS masks multiple whole sentences that are principle to the document and only generate the masked sentences as a single output sequence. We use the PEGASUS-LARGE[4] model available from the *huggingface* library. PEGASUS has 568 Million parameters, uses *Sentencepiece-Unigram* tokenization, and its pretraining objective is *Gap Sentence Generation*. The vocabulary size of this PLM ($|V_{\text{PLM}}|$) is 96103.

A.2 Datasets

We describe here three details on the target task dataset mentioned briefly in Section 4.1: (i) a detailed description of target task datasets, (ii) training-validation-test data splits, and (iii)the training data cleaning procedure.

Target Task Dataset Details

We use four target task datasets in this study: two query-focussed summarization dataset: EBM and BioASQ and two recent benchmark medical question summarization datasets: MeQSum and CHQSum each of which we describe below.

- **EBM** [Mollá and Santiago-Martínez, 2011]. Here input to the system is a query along with a PubMed abstract, and the expected output is the summary answering the question with the PubMed Abstract as the context.
- **BioASQ** [Tsatsaronis *et al.*, 2015]. We use the dataset from BioASQ-9B Phase-B summarization task. The input to the system is a question followed by relevant snippets from a collection of PubMed Abstracts. There are two kinds of outputs an exact answer and an ideal answer associated with the input. For the summarization task, we consider the ideal answer as the Reference summary.
- **MeQSum** [Ben Abacha and Demner-Fushman, 2019]. The dataset is created for better medical question summarization because the original patients' questions are verbose. The dataset contains 1000 patients' health questions selected from a collection distributed by the U.S. National Library of Medicine. Each question is annotated with a summarized question by medical experts.
- **CHQSum** [Yadav *et al.*, 2022]. CHQSum consists of 1507 domain-expert annotated question-summary pairs from the Yahoo community question answering forum[5] which provides community question answering threads containing users' questions on multiple diverse topics and the answers submitted by other users. The authors with the help of 6 domain experts identified valid medical question from the forum and asked the experts to formulate an abstractive summary for the questions.

In case of EBM and BioASQ, for each data point we append the Query and the PubMed Abstracts to form the input SD for the summarization model, and RS is the answer of the datapoint. In case of MeQSum and CHQSum, for each datapoint SD is the healthcare question and RS is the expert annotated summary of the question.

Train-Validation-Test splits for Target Task Datasets

We used the pre-defined train-valid-test split CHQSum provided in the original studies. BioASQ provided only train-test split thus we further split train set into train and validation set. Since the train-validation-test splits for MeQSum and EBM were not provided, we shuffled the dataset and considered a 70-15-15 data split.

Data Cleaning details

We perform two data cleaning steps on the training splits of EBM and BioASQ. We remove those data points for which there is (a) no medical-concept-bearing words overlap between RS and SD, and (b) length of RS is more than length of SD. From the EBM dataset having the original size of train split as 1483, the first step filters out 35 data points, and the second step filters out 25 data points, thus resulting in final training set size of 1423 data points. From BioASQ having the original size of train split as 2420, first step filters out 0 data points and second step filters out 805 data points, resulting in final train split size of 1525.

A.3 Evaluation Metrics

We first describe the implementation details for computing Rouge scores discussed in Section 4.2, where we use the official Rouge [Lin, 2004] script[6]. The following parameters: *-c 95 -2 -1 -U -r 1000 -n 4 -w 1.2 -a*, are used and we report the median at a 95% confidence interval.

However, Rouge fails to match words differing in surface form but belonging to the same medical concept. We thus

[2]https://github.com/nlpyang/PreSumm

[3]https://huggingface.co/facebook/bart-large

[4]https://huggingface.co/google/pegasus-large

[5]https://webscope.sandbox.yahoo.com/catalog.php?datatype=l&did=11

[6]https://github.com/bheinzerling/pyrouge/tree/master

develop **MedRouge** (MR) evaluation metric that adds a medical concept normalization step using QuickUMLS [Soldaini and Goharian, 2016] over Rouge; MR-1 and MR-2 are the two variants corresponding to Rouge-1 (R-1) and Rouge-2 (R-2). First, we mark unigrams (for Rouge-1) and bigrams (for Rouge-2) in the generated summaries that have surface form overlap with the reference summaries. Next, from the remaining set of unigrams and bigrams, we identify those with no surface form overlap but overlap at concept level (e.g., *treatment* and *therapy*) and flag them to consider for the final Rouge-1 and Rouge-2 computation.

A.4 Baseline Models

Here, we provide implementation details regarding the vocabulary adaptation and intermediate fine-tuning baseline models as described in Section 4.3.

- **IFT-CNN**: In this setup, we take the PLMs models and do the intermediate fine-tuning using CNN/DM news summarization dataset. This acts as a strong baseline against MEDVOC.
- **IFT-PAC**: In this setup, we use the PubMed Abstract Collections (PAC) dataset to perform the task of intermediate fine-tuning. This model is MEDVOC without vocabulary adaptation.
- **BSA-BioBERT**: BioBERT [Lee *et al.*, 2019] continuously pre-train *bert-base-cased* model on a huge biomedical corpora containing PubMed abstracts, PMC-Articles. We replace the encoder component of the BertSumAbs model which was previously *bert-base-uncased*, with the BioBERT model in this case.
- **BSA-PubMedBERT**: Unlike BioBERT, PubMedBert [Gu *et al.*, 2021] learns the model vocabulary from scratch and is vastly different from the original Bert vocabulary. Similar to BSA-BioBERT, we replace the encoder component of the BertSumAbs model, with the PubMedBERT model.
- **BSA-AVoCado**: AVocaDo [Hong *et al.*, 2021] is a work primarily in the classification field. For fair comparison, we only incorporate the vocabulary adaptation module where the subwords to be added to the V_{PLM} vocabulary is identified. We then use this updated vocabulary and perform IFT-PAC using PAC as we do in the case of MEDVOC for BertSumAbs.

A.5 Hyperparameters

We discuss the following hyperparameters in line with discussion in Section 4.4 as follows: (i) the optimal hyperparameters obtained for each target task dataset using MEDVOC, (ii) the training hyperparameters, (iii) inference hyperameters.

Hyperparameter Search for Vocabulary Construction

During the vocabulary construction for MEDVOC as described in Algorithm 1, we identified two hyperparameters K and A that we obtain for each of the models for each of the target task datasets. We report the optimal values thus obtained in Table 5. We observe that these values vary drastically across different model types.

Dataset	BSA				BART				PEGASUS			
	V_{PLM}: 30522				V_{PLM}: 50265				V_{PLM}: 96103			
	K	A	$\lvert V_{\text{MEDVOC}}\rvert$	Time	K	A	$\lvert V_{\text{MEDVOC}}\rvert$	Time	K	A	$\lvert V_{\text{MEDVOC}}\rvert$	Time
EBM	15K	0.25	32121	2	15K	8	61326	8	20K	10	97621	14
BioASQ	15K	0.25	32941	3	15K	5	56727	8	20K	10	98721	14
MeQSum	15K	0.25	30689	1	10K	8	51012	8	15K	1	96133	9
CHQSum	15K	0.2	30695	1	10K	7	50945	8	20K	1	96117	7

Table 5: The optimal hyperparameter values of Algorithm 1. We then provide the resultant MEDVOC vocabulary size and the time required for hyperparameter search for each target dataset and PLM model type in hours.

Training Hyperparameters

All the experiments are run on three 32GB Tesla V100 GPUs. The final training times for IFT-PAC for MEDVOC is mentioned in Table 6. We descibe two types of hyperparameters.

- **Common Hyperparameters:** *number of epochs* for fine-tuning: 5, *check-pointing*: 2500 steps, and *accumulation steps*: 10 (for BertSumAbs) and 2 (for BART and PEGASUS).
- **PLM-specific Hyperparameters:** In case of BertSumAbs there are four additional hyperparameters: learning rate for encoder: 0.002, learning rate for decoder: 0.01, warm-up steps for encoder: 20000, warm-up steps for decoder: 15000. In case of BART and PEGASUS, we use the huggingface scripts[7] and its associated default hyperparameters, like learning rate: 5e-5 and modify source and target length based on target task datasets for appropriate input truncation to do the fine-tuning.

Dataset	BSA	BART	PEGASUS
EBM	30	30	86
BioASQ	36	34	78
MeQSum	30	28	82
CHQSum	27	28	81

Table 6: Time required in hours for intermediate fine-tuning using PAC for each target task dataset and PLM model setting.

Inference Hyperparameters

We used beam search to run the **inference** on the test set. We tuned the following hyperparameters of beam search: beam size (B $\in [2, 10]$) and length-penalty [Wu *et al.*, 2016] ($lp \in (0.1, 3]$) on the validation split of the target task dataset. The best values of hyperparameters thus obtained are mentioned in Table 7.

Dataset	BSA		BART		PEGASUS	
	B	lp	B	lp	B	lp
EBM	7	2.5	2	0.7	9	2.5
BioASQ	7	2.5	8	3	9	3.5
MeQSum	8	0.7	6	0.3	8	0.9
CHQSum	6	0.6	6	0.3	8	0.9

Table 7: Optimal values for inference hyperparameters - beam size (B) and Length Penalty (lp) used for beam-search generation for each of the PLM against each of the datasets.

[7] https://github.com/huggingface/transformers/tree/main/examples/pytorch/summarization

B Experimental Results

Here we describe two things: (i) additional evaluation of MEDVOC in extenstion to discussion in Section 5.1, and (ii) human evaluation setup as discussed in Section 5.3.

B.1 Additional Evaluation of MEDVOC

Here we provide a discussion of MEDVOC along three dimensions: (i) how does MEDVOC perform against standard vocabulary adaptation baselines like AVocaDo in terms of fragment score, (ii) performance comparison using MedRouge, and (iii) comparison with three LLMs.

MEDVOC results in better fragment score than AVocaDo

We compare the variation in fragment scores observed in MEDVOC and BSA-AVocaDo. In MeQSum and CHQSum, the resultant vocabulary size from AVocaDo is more than that of MEDVOC, and for EBM and BioASQ the trend is reversed. To understand the effect of vocabulary adaptation and for a fair comparison across differing vocabulary sizes, we first remove the common part from both the vocabularies and make vocabularies of the same size either by randomly sampling or selecting top elements from the larger vocabulary. We find that in all the cases MEDVOC results in a lower (or comparable) fragment score to that of AVocaDo. We report the original fragment scores and the best fragment scores obtained from AVocaDo and MEDVOC using the strategy discussed above in Table 8. This also ascertains our hypothesis of the correlation between optimizing fragment score on downstream target task dataset and target task performance.

Dataset	Original		Overlap Removed	
	AVocaDo	MEDVOC	AVocaDo	MEDVOC
EBM	1.53	1.51	1.55	1.53
BioASQ	1.62	1.61	1.64	1.62
MeQSum	1.40	1.38	1.41	1.41
CHQSum	1.28	1.27	1.30	1.29

Table 8: Fragment score values observed for AVocaDo and MEDVOC for BertSumAbs model on three datasets. **Original** block refers to the fragment score obtained from the original vocabulary of AVocaDo and MEDVOC. In **Overlap Removed** block, we remove the overlapping vocabulary and use an equal-sized vocabulary. We see that in 7 settings MEDVOC results in a 1.07% lower fragment score on average than AVocaDo.

MEDVOC outperforms baselines in terms of MedRouge

We observe that the same performance improvement trend as seen with Rouge-L also holds for MedRouge (Table 9). There is a performance increase in terms of MedRouge (MR-1 and MR-2) metric of 6.99%, 10.88%, 7.91%, and 1.36% performance improvement over baselines across EBM, BioASQ, MeQSum, and CHQSum datasets respectively. This indicates that even if there is a surface-level mismatch, a major portion of the summaries are meaningful and match at the concept level.

MEDVOC performs at par with LLMs

We observe from Table 10 that MEDVOC outperforms standard LLM models such as LLaMa-7B, LLaMa-13B, and MedAlpaca, by a statistically significant margin of 54.32% in terms of Rouge-L. We follow the work of Jahan et al. (published in arXiv on October 6, 2023) to use the prompt for the medical question summarization datasets of MeQSum and CHQSum to use the prompt: "*Rewrite the following question in a short and concise form*". In the case of medical document summarization datasets of BioASQ and EBM, we use the prompt "*Summarize the following paragraph in less than 200 words*". We evaluate LLMs in two settings: zero-shot and few-shot (one-shot in-context learning). In both the settings, LLaMa-7B achieves the best performance whereas MEDVOC is second in few-shot comparision. In full dataset setting, MEDVOC beats all the LLMs in zero-shot setting by a margin of 54.32% .

Model	BSA		BART		PEGASUS	
	MR-1	MR-2	MR-1	MR-2	MR-1	MR-2
EBMSumm						
SOTA	27.00	10.17	27.02	13.21	26.96	9.96
IFT-CNN	28.37	10.04	29.50	11.12	26.96	9.96
IFT-PAC	29.73	10.14	29.80	11.16	28.71	11.26
MEDVOC	30.09	11.50	31.07	11.23	31.00	11.86
BioASQ						
SOTA	50.71	40.55	53.15	43.02	46.72	33.25
IFT-CNN	46.30	36.00	49.42	40.30	46.72	33.25
IFT-PAC	51.31	41.24	51.69	41.09	46.68	37.67
MEDVOC	53.75	43.55	53.94	42.56	48.79	36.25
MeQSum						
SOTA	57.28	44.82	59.94	46.93	58.25	45.33
IFT-CNN	50.71	35.38	63.82	50.18	58.25	45.33
IFT-PAC	54.50	40.58	63.44	49.60	63.12	50.91
MEDVOC	59.01	44.88	62.34	49.26	60.41	47.45
CHQSum						
SOTA	41.67	22.93	46.36	28.44	47.86	30.34
IFT-CNN	42.54	24.20	47.05	29.53	47.86	30.34
IFT-PAC	43.78	25.03	46.94	29.52	48.04	30.24
MEDVOC	44.14	24.90	47.77	29.30	48.10	30.41

Table 9: Median MedRouge values for MEDVOC, baselines and SOTA models. We find that MEDVOC outperforms baselines and even SOTA in majority of the cases.

Model	EBM	BioASQ	MeQSum	CHQSum
Zero-Shot				
LLaMa-7B	16.78	29.03	34.78	28.69
LLaMa-13B	16.55	28.06	29.26	23.52
MedAlpaca	13.90	25.20	31.67	24.83
MEDVOC	15.00	27.10	29.60	25.20
Few-Shot				
LLaMa-7B	16.39	21.62	37.90	32.61
LLaMa-13B	15.89	20.13	20.10	23.22
MedAlpaca	13.74	20.93	36.44	28.41
MEDVOC	16.10	30.40	37.70	28.45
Full				
MEDVOC	20.03	45.98	53.63	38.75

Table 10: Rouge-L values obtained using zero-shot and few-shot (one-shot In-Context Learning) inference from LLMs. MEDVOC values are averaged across the three PLMs. MEDVOC trained on a full dataset beats the zero-shot performance of LLMs, and MEDVOC's zero-shot performance achieves the second-best rank as LLaMa-7B achieves the best Rouge-L score.

B.2 Human Evaluation

The annotations were conducted on the Prolific[8] platform using 30 participants in total. Each participant was shown 10 random samples from a pool of 100 summary pairs (order of summary randomized and anonymized) and was given 30 minutes to complete the study. We also collected demographic information and annotation experience feedback from the participants. The study instrument was designed using the Potato tool [9]. The participants were compensated at the rate of 9 pounds/hour, which according to platform guidelines was fair compensation. Proper consent notice was shown to the participants and no personal information (other than age) was collected in the demographic information. The filtering criteria for participants were kept as follows:

- **Age:** ≥ 25,
- **Primary Language:** English,
- **Highest education level completed:** Graduate degree (MA/MSc/MPhil/other), Doctorate degree (PhD/other)
- **Subject:** Medicine, Health and Medicine, Biomedical Sciences.

The annotations were carried across three dimensions [Fabbri *et al.*, 2021b] each of which we discuss below.

- **Coherence**. The summary should be well-structured and well-organized. The summary should not just be a heap of related information but should build from sentence to sentence to a coherent body of information.
- **Relevance**. The summary should include only important information from the source document. In the case of a query, you must also judge how relevant is the summary to the query based on the given source document.
- **Faithfulness**. A faithful summary contains only statements that are entailed by the source document. You may also penalize summaries that contain facts not supported (or can not be verified) in the source document (termed as hallucinated facts).

For each of these dimensions, we show one positive (high rating) and one negative example (low rating) along with an explanation as a part of our annotation guideline (Table 11).

Demographic analysis of participants. The average age of participants was 30 years. Out of 30 participants, 70% were female and 30% were male. 90% were Graduates, and 10% were PhD holders. 60% of the participants were practicing medicine, or previously practiced in a clinical setting. The participants came from 11 different countries, majority of which came from UK (n=11) and US (n=4).

Annotation experience of participants. After the study, we took feedback from the participants regarding the overall annotation experience. Our primary focus was on three aspects: (i) Clarification of the annotation guidelines, (ii) Usability of Potato, (iii) Overall experience with the study setup. 60% of participants found the annotation guidelines to be mostly clear, and 30% found it very clear. 80% of the participants were satisfied with the Potato platform interface. 90% of the participants were overall satisfied with the study design and summary pairs shown to them.

[8] https://www.prolific.com/

[9] https://github.com/davidjurgens/potato

Source Document	KNEE OSTEOARTHRITIS. Good morning about 20 years ago I suffered ruptured anterior cruciate ligament and removal of domestic law meniscus, was operated and made me clancy, at present unfortunately my knee is totally affected and I have arthritis and severe pain, according to a dr traumatologo commented me I need a knee prosthesis my question is can you treat me, or turn me can recommend doctors or hospitals to treat in the U.S.
	Positive Example
Summary	How can I find physician(s) or hospital(s) who specialize in knee osteoarthritis?
Rating	5
Explanation	Here we can see the summary is focused on knee osteoarthritis and asks how to find physicians or hospitals who specialize in it.
	Negative Example
Summary	What are the treatments for ruptured anterior cruciate ligament and meniscus?
Rating	2
Explanation	In the source document the patient is asking for recommendations for hospitals or doctors who specialize in the treatment for the topic of knee osteoarthritis.

Table 11: A negative and positive example as shown to the participant in the annotation guidelines for clarification under *Relevance* dimension of annotation. The data point is taken from MeQSum dataset.

C Limitations

First, the evaluation of MEDVOC in this work is limited only to encoder-decoder-based PLMs, we thus plan to extend and evaluate MEDVOC on decoder-only-based PLMs like GPT. Second, in MEDVOC we identify medical-concept-bearing using the QuickUMLS tool which uses certain heuristics to identify such words. Although we maintain a very high threshold, there will always be some errors that might cascade through the entire pipeline. Third, we observe that in the case of MEDVOC although the speedup achieved is very high, owing to the nature of efficient hyperparameter search. However, the drawback here is that we have to perform one iteration of IFT per target task dataset which is still a huge cost and we will try to alleviate this problem in future iterations. Fourth, although the vocabulary adaptation pipeline of MEDVOC is quite flexible to adapt to any domain with high vocabulary mismatch, we do not evaluate the generalizability of MEDVOC over non-medical domains such as legal text and scientific literature of rarer subjects.

D Ethics Statement and Broader Impact

Summarization systems, in general, any generation systems are prone to hallucination [Ji *et al.*, 2022] leading to the generation of less faithful summaries. Furthermore, our human evaluation pointed out that MEDVOC generates significantly more fraction of faithful summaries when compared to existing baselines. Our view is that summaries produced by such PLMs are not yet production-ready for their intended users like medical professionals, and clinicians. More thorough research is needed to better characterize the kinds of errors (specifically in the context of faithfulness and relevance) made by these PLMs and ultimately to mitigate them.