Generating Distributable Surrogate Corpus for Medical Multi-label Classification

Seiji Shimizu, Shuntaro Yada, Shoko Wakamiya, Eiji Aramaki
Nara Institute of Science and Technology, Nara, Japan
{shimizu.seiji, s-yada, wakamiya, aramaki}@is.naist.jp

Abstract

In medical and social media domains, annotated corpora are often hard to distribute due to copyrights and privacy issues. To overcome this situation, we propose a new method to generate a surrogate corpus for a downstream task by using a text generation model. We chose a medical multi-label classification task, *MedWeb*, in which patient-generated short messages express multiple symptoms. We first fine-tuned text generation models with different prompting designs on the original corpus to obtain synthetic versions of that corpus. To assess the viability of the generated corpora for the downstream task, we compared the performance of multi-label classification models trained either on the original or the surrogate corpora. The results and the error analysis showed the difficulty of generating surrogate corpus in multi-label settings, suggesting text generation under complex conditions is not trivial. On the other hand, our experiment demonstrates that the generated corpus with a sentinel-based prompting is comparatively viable in a single-label (multiclass) classification setting.

Keywords: Text Generation, Language Model, Privacy Protection, Social Media

1. Introduction

Supervised machine learning, which is the de facto standard in today's natural language processing (NLP), requires annotated corpora. Although sharing corpora with researchers enhances further development in scale, annotated corpora may not be distributed due to privacy policies and copyrights. Especially in the medical domain, this problem arises frequently and critically (Hahn and Oleynik, 2020; Aramaki et al., 2022). Also, social media posts may not only contain some personal information but are also often limited to content-excluding distribution in the platform's terms[1].

Two major approaches have been taken to tackle the problem of difficulty in corpus distribution. The first approach is to delete personal information in the corpus, that is, de-identification (Sibanda and Uzuner, 2006; Uzuner et al., 2007) or anonymization (Zuo et al., 2021), which is well studied in the medical domain. MIMIC (Johnson et al., 2016) is the most popular de-identified corpus in the medical domain. However, it is costly and difficult to achieve perfect de-identification of arbitrarily large corpora, regardless of whether the method is based on machine learning or human labor.

The second approach is to generate new corpora in which any *real* person's information is not contained. One such corpus is *MedWeb* (Wakamiya et al., 2019), where patients' self-reports of symptoms were composed manually via crowdsourcing. Whereas manually generating data is highly costly, model-based automatic generation enables large-scale and low-cost corpus creation. The recent advance in text generation (Zhang et al., 2022) promotes such an approach, for example, in the social media domain (Claveau et al., 2021) and in the medical domain (Amin-Nejad et al., 2020). However, existing studies investigate the viability of such generated corpora mainly for data augmentation, which extends the existing *small* datasets to be larger for data-hungry deep learning models. The generated corpora in this approach are to be mainly combined with the original dataset. The remaining question is: *Can a synthetic corpus created by text generation be a surrogate for a downstream task?*

This study aims at generating a distributable surrogate corpus and investigating its viability in the downstream task. We set the downstream task to multi-label classification in the medical domain, i.e., the aforementioned *MedWeb* task: Multiple symptoms (such as runny nose and cough) expressed in patient-generated short messages must be correctly labeled. We first generate synthetic corpora by generation models trained on the original corpus. Then, we evaluate the quality of the generated corpora by solving the task with classification models.

Specifically, in the generation step, we fine-tuned text generation models with different prompting methods (i.e., the sentinel tokens and soft prompts) to obtain different qualities of generated corpora. In the evaluation (classification) step, we trained the classification models on either the generated corpora or the original corpus. The flow of this experiment is outlined in Figure 1.

While a few recent studies (Claveau et al., 2021; Amin-Nejad et al., 2020; Ive et al., 2020) started investigating the viability of generated corpora as a replacement for the original datasets, we tackle

[1] https://twitter.com/en/privacy

20 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

the following challenging settings:

User-generated text: Our target corpus to generate is patient-generated text, which depends highly on context. The textual nature, thus, becomes ungrammatical and fragmented.

Multi-label condition: A patient-generated message of the target corpus has multiple symptom labels. The generation model must understand the multiple conditions to create a correct message that expresses the corresponding symptoms.

The contributions of this paper are as follows:

- We propose a text generation approach for **patient-generated corpus** using pre-trained text generation models.
- We evaluated the proposed approach using an existing dataset in a **multi-label classification task** in the medical domain (that is, *MedWeb*).

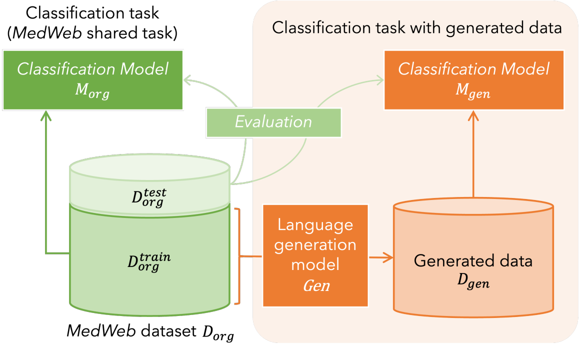

Figure 1: Flow of our experiment. Language generation models (Gens) were trained using the original *MedWeb* training dataset D^{train}_{org} with different prompting methods. Classification models (M_{org} and M_{gen}s) are trained on the *MedWeb* training data D^{train}_{org} or the generated corpora (D_{gen}s), respectively. The *MedWeb* test data D^{test}_{org} is used for evaluation of both M_{org} and M_{gen}.

2. Related Work

2.1. Training Corpus Generation

Most studies on corpus generation are motivated by data augmentation rather than the creation of surrogate corpus. On the other hand, this study aims to replace the original corpus. Some studies evaluated surrogate corpus as part of their experiments in the medical and social media domains.

2.1.1. Medical Domain

A few previous studies utilized a pre-trained language model (PLM) for text data generation in the medical domain. The generated text was used as a training corpus.

Amin-Nejad et al. (2020) utilized PLM for Electric Health Record (EHR) generation. They generated texts given the patient's conditions, including demographic data, diagnosis, procedures, medications, microbiology tests, and laboratory tests. Generated data were evaluated based on the performance of unplanned readmission prediction and phenotype classification. Generated data accomplished comparable results with original data. In addition, this study showed that when combined with original data, using generated data improves the performance of classifiers in downstream tasks.

One of the advantages of automatic text generation is that it can generate a large number of text that are hard to sample from the real world. Motivated by the lack of data for rare symptoms, PLM was used for the generation of symptom definitions alongside with biomedical dictionary in Kim and Nakashole (2022). Given one symptom or two symptoms, definitions were generated. Generated definitions were used in vaccine side effect detection.

Pappas et al. (2022) also applied a pre-trained language generation model for data augmentation. They experimented with different data augmentation approaches for biomedical factoid question answering. As one of the approaches, they utilized question generation using fine-tuned T5 (Raffel et al., 2020). ALBERT (Lan et al., 2019) was used in the downstream task (biomedical factoid question answering). They found that adding generated data to original data results in slightly better performance than only using original data.

2.1.2. Social Media Domain

The generation of social media posts can also be important because it also cannot be distributed for confidentiality reasons. Claveau et al. (2021) utilized a pre-trained language generation model (GPT-2) (Radford et al., 2019) for a surrogate training corpus generation. In downstream tasks, namely sentiment analysis on product reviews and fake news detection, the quality of generated corpora was evaluated. In neural classification approaches, they trained BERT (Devlin et al., 2019) as a classification model using 1) only original corpus, 2) only generated corpus, and 3) a mixture of original corpus and generated corpus. As a result, they found that 1) leads to better classifier performance than 2) and without filtering, 1) tends to perform better than 3).

2.2. Controllable Text Generation

Controllable text generation (CTG) is one of the hottest research topics in recent NLP. There could be many possible ways to achieve a patient-generated training corpus.

Zhang et al. (2022) gave a comprehensive survey on different approaches for CTG, which are 1) fine-tuning, 2) retrain/refactor PLMs and 3) post-process. Also, as described in Zhang et al. (2022), there are three major categories of fine-tuning approaches: prompt-based approaches, adapted module approaches, and reinforcement learning inspired approaches.

Jiang et al. (2021) showed that the performance of PLM is sensitive to prompt design modification. Liu et al. (2021a) provided a survey on different prompting approaches and those performances. In terms of methods for designing prompts and their human effort, most of the methods can be categorized into two: hand-crafted and automated search. Hand-crafted is the approach where humans design the suboptimal suitable prompt, while automated search is the approach where a suitable prompt is chosen automatically.

Among automated search approaches, Liu et al. (2021b) and Lester et al. (2021) experimented with soft prompts, tokens with trainable embeddings introduced in the fine-tuning stage. By inserting soft prompts, PLM can automatically search for an optimal prompt in the continuous space of all possible prompts. Wang et al. (2022) applied the soft prompt method for data augmentation in few-shot settings. Generated corpora were used for sequence labeling and sentence classification tasks.

T5 (Raffel et al., 2020) was used in corpus generation and BERT was used for downstream tasks. They found that adding soft prompts is effective in both downstream tasks. Also, Lester et al. (2021) experimented with sentinel tokens used in pre-training of T5. In pre-training of T5, unique sentinel tokens are used for marking masked spans in the input text. The task for T5 is to reconstruct these masked spans. They showed that in some experimental settings, using sentinel tokens in prompts is beneficial for the performance of PLM.

In the context of natural language generation, Schick and Schütze (2021) experimented with combining multiple instructions (prompts) through knowledge distillation. They evaluated the proposed automated search in a summarization task.

Although most prompt designing methods mentioned above are proposed in the context of Natural Language Understanding (NLU) tasks, we applied those methods to experiment with different prompting methods.

3. Dataset

MedWeb dataset consists of pseudo-posts for multi-label symptom classification.

To avoid privacy issues, the dataset was developed by crowdsourcing and not extracted from the actual X (previously Twitter) platform. In crowdsourcing, data were constructed from the symptom keywords (called "seed words") that frequently appeared in real-world disease-related posts. Each post includes a description of whether the X (previously Twitter) user is experiencing a combination of symptoms, that is, a combination from a set "*Influenza*", "*Diarrhea*", "*Hay fever*", "*Cough*", "*Headache*", "*Fever*", "*Runny nose*", and "*Cold*". Examples of pseudo-posts in the dataset are listed in Table 1.

Classifying a combination of symptoms given a post can be considered a multi-label classification task among NLP tasks. In the following sections, we refer to one label pattern as a symptom combination.

MedWeb dataset contains 2,560 posts, and the ratio of training to test data is 75% (1,920 posts) to 25% (640 posts). "*No symptom*", "*Cold*", "*Runny nose*", "*Fever*", "*Headache*", "*Cough*", "*Cold and Runny nose*", "*Hay fever and Runny nose*", "*Diarrhea*", and "*Influenza and Fever*" consist of 1,754 posts in total, which is 91% of all *MedWeb* training data.

4. Method

To investigate the viability of automatically generated corpora as surrogate training datasets, we trained; 1) the generation model (Section 4.1), which is utilized to create generated data (denoted with D_{gen}) and 2) the classification models (Section 4.2), which are used to evaluate the quality of the generated texts.

4.1. Generation Models

We fine-tuned a text generation model Gen on the original *MedWeb* training dataset D_{org}^{train}. Following the previous study (Amin-Nejad et al., 2020), we decided to choose a fine-tuning approach among different controllable text generation (CTG) approaches. We used T5 (Raffel et al., 2020) as Gen in our method. Specifically, we fine-tuned the model pre-trained on Japanese corpora.[2] Since, we are handling a data-to-text generation task, encoder-decoder-type models are suitable. We considered T5 as representative of such models.

Gen is fine-tuned in the following manner: given a symptom combination, it should generate a post

[2] https://huggingface.co/sonoisa/t5-base-japanese

Post	*Influenza*	*Diarrhea*	*Hay fever*	*Cough*	*Headache*	*Fever*	*Runny nose*	*Cold*
風邪をひくと全身がだるくなる。 (The cold makes my whole body weak.)	–	–	–	–	–	–	–	+
花粉症の症状が出てたのは久しぶりだ。 (It's been a while since I've had allergy symptoms.)	–	–	+	–	–	–	+	–
インフルエンザのワクチン打ちに行ってきた。 (I went to get vaccinated for the flu.)	–	–	–	–	–	–	–	–

Table 1: Examples of *MedWeb* pseudo-posts. English translations are shown in the examples. + sign stands for the existence of the corresponding symptom in the user; – sign stands for the absence.

that expresses the corresponding symptoms. For example, when the model is given a combination of "*fever and headache*", the generated post should say, for example, "*I had a fever today. Bad headache too...*". Among various ways to achieve this conditional text generation, we chose prompting as a method because of its conceptual simplicity and relative efficiency in computational cost. Previous studies (Lester et al. (2021), Jiang et al. (2021), Liu et al. (2021b)) showed that giving appropriate instruction improves the performance of large generative PLMs on multiple tasks. Based on those work, we chose the following prompting methods:

BASE (bs): A hand-crafted symptom prompt.

As a baseline prompting method, we designed hand-crafted prompts. We represented a combination of symptoms by symptom name + the description of whether the symptom should appear or not. We put this expression at the beginning of the input sentence. To transform the instruction into the form of a question, we put "のTweetは？ (What is the tweet?)" at the end of the input sentence.

SENTINEL (st): A hand-crafted symptom prompt with a sentinel token.

We added a sentinel token (denoted with <X>) to the BASE prompt. Adding the sentinel token makes the task more similar to the task in pre-training of generative PLMs. We expected that catastrophic forgetting of the model could be avoided by making fine-tuning stage more similar to pre-training.

SOFT (sf): A hand-crafted symptom prompt with soft prompt tokens.

In our baseline design, we added soft prompt tokens (<s[id]>) that are trained simultaneously with the model parameters, inspired by Liu et al. (2021b) and Lester et al. (2021). This method was originally adopted to solve natural language understanding tasks. We assumed that this method would work for text generation too.

SENTINEL+SOFT (st+sf): A hand-crafted symptom prompt with soft prompts and a sentinel token.

We applied two modifications (the sentinel token and soft prompts addition) to the baseline design.

Examples of the four prompt designs given the same symptom combination are listed in Table 2. Finally, we created four Gens and 40 D_{gen}s (10 datasets per prompt design) as we will explain in Section 5.1.

4.2. Classification Models

We evaluate the D_{gen} quality on a classification task, the same as the *MedWeb* shared task.

We compared the evaluation results of different models using D_{org}^{test}. To do so, we trained the classification models using data created by different Gens (mentioned in Section 4.1). We also trained the classification model using D_{org}^{train}.

We trained Gens using the prompt BASE, SENTINEL, SOFT, and SENTINEL+SOFT, and generated posts (D_{gen}s) from each Gen. Then we trained individual classification models on the different D_{gen}s. $M_{gen(bs)}$, $M_{gen(st)}$, $M_{gen(sf)}$, and $M_{gen(st+sf)}$ respectively denote these classification models.

The classification models used in our experiment are a pre-trained BERT model[3] with a linear transformation layer for the symptom combination classification. We trained our classification models on the task of symptom combination classification. Given a post, the model estimates the probabilities for eight symptom labels. When the output probability of a label surpassed a threshold, we considered the post to contain the corresponding symptom.

5. Experimental Setup

We evaluated the generated posts on D_{org}^{test} by measuring the performance of the different M_{gen}s compared to M_{org}. The flow of this experiment is shown in Figure 1.

The hyperparameters for the models M_{gen} and M_{org} are as follows: M_{gen} is trained for 20 epochs with a batch size of 32 using the Adam optimizer with 3e-4 learning rate, while $M_{org}$is trained for 24

[3] https://huggingface.co/cl-tohoku/bert-base-japanese

Prompt design	Example prompt
BASE	インフルエンザの症状なし、...鼻水・鼻づまりの症状あり、風邪の症状なしのTweetは？ (What is the tweet in which the symptom of influenza doesn't appear, ...runny nose appears, and cold doesn't appear?)
SENTINEL	インフルエンザの症状なし、...鼻水・鼻づまりの症状あり、風邪の症状なしのTweetは？<X> (What is the tweet in which the symptom of influenza doesn't appear, ...runny nose appears, and cold doesn't appear?<X>)
SOFT	インフルエンザの症状なし、...鼻水・鼻づまりの症状あり、風邪の症状なしのTweetは？<s1>...<s99> (What is the tweet in which the symptom of influenza doesn't appear, ...runny nose appears, and cold doesn't appear?<s1>...<s99>)
SENTINEL+SOFT	インフルエンザの症状なし、...鼻水・鼻づまりの症状あり、風邪の症状なしのTweetは？<s1>...<s99><X> (What is the tweet in which the symptom of influenza doesn't appear, ...runny nose appears, and cold doesn't appear?<s1>...<s99><X>)

Table 2: Examples of different prompt designs, given "*Runny nose*" as the input symptom combination. `<X>` denotes the sentinel token and `<s[id]>` denotes soft prompt tokens.

epochs with a batch size of 8 using the AdamW optimizer with 1e-8 learning rate. As for the generation process, the hyperparameters include setting the number of beam search and beam groups equal to the number of posts for each label, a diversity penalty of 0.4, and a temperature value of 1.0.

5.1. Generation of Data

In order for the generated dataset to be distributable and comparable to D_{org}^{train}, D_{gen} should meet the following conditions. 1) generated post should be de-identified and 2) the distribution of symptom combinations should be the same as that in D_{org}^{train}. To ensure that the settings of the task are the same for $M_{gen}s$ and M_{org}, we made the distribution of conditional labels for a generation the same as that of the distribution of labels in the D_{org}^{train}. Because of these conditions, additional steps were needed to generate posts. Using the fine-tuned Gens, we first generated a larger number of posts for each symptom combination than that of D_{org}[4]. Then, we subtracted the subset of generated posts using an exact match search. The ratios of exactly matched posts generated by Gens using the prompt BASE, SENTINEL, SOFT, and SENTINEL+SOFT were 2.0%, 1.5%, 1.1%, and 1.1%, respectively[5].

5.2. Evaluation

We evaluated classification models using the basic metrics used in the *MedWeb* shared task, and those are precision (micro average), recall (micro average), F1 score (micro average) and exact match accuracy. D_{org}^{test} is used for the evaluation of classification models. We created D_{gen} for each prompt 10 times and trained 10 M_{gen} for each D_{gen}, resulting in 100 models with different parameters per prompt. Similarly, we trained M_{org} 10 times on D_{org}^{train}. We obtained the scores (precision, recall, F1 score, and exact match accuracy) for each 100 models for M_{gen} and 10 models for M_{org}, and present the average scores as the scores for $M_{gen(bs)}$, $M_{gen(st)}$, $M_{gen(sf)}$, $M_{gen(st+sf)}$, and M_{org}.

We evaluated the performance in the following three settings:

- Multi-symptom setting which is a usual multi-label classification ($multi$ in short).
- Single-symptom classification using all data including multi-symptom posts for training data ($single_{all}$ in short).
- Single-symptom classification without using multi-symptom posts for training data ($single_{only}$ in short).

5.3. Manual Evaluation of Fluency

Additionally, we independently evaluated the fluency of the generated corpus with a Turing-test-like evaluation. To do so, we built the mixed 300 test set, which consisted of 150 generated posts and 150 original posts. We asked three Japanese annotators (two of them are nurses with more than 10 years of experience) to label the constructed data. Given a post, the annotators labeled whether the post is from original data or generated data.

6. Results

6.1. Results of Classification

The results are summarized in Tables 3, 4 and 5. Multiple M_{gen} with different prompting methods are denoted with $M_{gen(bs)}$ (prompted by BASE), $M_{gen(st)}$ (prompted by SENTINEL), $M_{gen(sf)}$ (prompted by SOFT), and $M_{gen(st+sf)}$ (prompted by SENTINEL+SOFT).

The results of $multi$ are shown in Table 3. Among the four prompting methods, $M_{gen(st)}$ gave the best result in terms of precision (0.757) and

[4] Since we have no a priori knowledge about the number of exactly matched posts to be generated, we generated 35% more posts for each symptom combination and then randomly sampled them.

[5] D_{gen} generated in this experiment is available at https://github.com/seiji-shimizu/medweb-gen

Model	Prompt	Accuracy	F1 (micro avg.)	Recall (micro avg.)	Precision (micro avg.)
$M_{gen(bs)}$	BASE	0.632 (0.0105)	**0.759** (0.0120)	**0.762 (0.0080)**	0.756 (0.0064)
$M_{gen(st)}$	SENTINEL	**0.654** (0.0105)	0.757 (0.0114)	0.758 (0.0085)	**0.757** (0.0065)
$M_{gen(sf)}$	SOFT	0.629 (0.0119)	0.716 (0.0095)	0.695 (0.0201)	0.738 (0.0120)
$M_{gen(st+sf)}$	SENTINEL+SOFT	0.615 (0.0062)	0.692 (0.0080)	0.671 (0.0176)	0.716 (0.0078)
M_{org}	-	0.855 (0.0325)	0.910 (0.0126)	0.919 (0.0317)	0.901 (0.0170)

Table 3: Scores for $multi$. Values in parentheses represent standard deviations of the scores from 10 models. The highest accuracy and F1 scores are presented in bold.

Model	Prompt	Accuracy	F1 (micro avg.)	Recall (micro avg.)	Precision (micro avg.)
$M_{gen(bs)}$	BASE	0.682 (0.0166)	0.800 (0.0171)	0.867 (0.0134)	0.742 (0.0148)
$M_{gen(st)}$	SENTINEL	**0.701** (0.0132)	**0.807** (0.0146)	**0.875** (0.0096)	**0.750** (0.0099)
$M_{gen(sf)}$	SOFT	0.677 (0.0120)	0.726 (0.0125)	0.741 (0.0260)	0.712 (0.0145)
$M_{gen(st+sf)}$	SENTINEL+SOFT	0.679 (0.0062)	0.673 (0.0115)	0.770 (0.0200)	0.718 (0.0065)
M_{org}	-	0.861 (0.0085)	0.915 (0.0172)	0.938 (0.0110)	0.889 (0.0074)

Table 4: Scores for $single_{all}$. Values in parentheses represent standard deviations of the scores from 10 models. The highest accuracy and F1 scores are presented in bold.

exact match accuracy (0.654). $M_{gen(bs)}$ gave the best F1 score (0.759) and recall (0.762). Since the highest exact match accuracy is the hardest to achieve, we consider $M_{gen(st)}$ to be the best performing M_{gen}. Compared to M_{org}, the best performing M_{gen} (that is, $M_{gen(st)}$) could not achieve comparable scores.

The results of $single_{all}$ are shown in Table 4. In this evaluation, we used the same classification models (trained on D_{gen} and D_{org}^{train}) as in $multi$ and excluded posts with multiple symptoms only from the test data. Both M_{org} and M_{gen} performed slightly better compared to the results of $multi$. The gap between M_{org} and the best performing M_{gen} was still relatively large.

The results of $single_{only}$ are shown in Table 5. We only evaluated $M_{gen(st)}$, which was the best performing M_{gen} model in other experiments. The gap between M_{org} and M_{gen} became smaller in this experiment. Compared with the results in Table 4, the scores of the best performing M_{gen} increased by 0.0407 on average. On the other hand, the scores for M_{org} increased by 0.0218 on average.

6.2. Results of Manual Evaluation of Fluency

The average accuracy of the labeling was 0.648 and average inter-human Cohen's kappa was 0.355. Both of those scores can be interpreted positively in the context of the Turing-test-like evaluation.

The low accuracy score suggests that the fluency of the generated corpus is relatively indistinguishable from that of the original corpus, and the task of labeling itself was difficult. Also, the low Cohen's kappa coefficient suggests the subjectivity of labeling. These results can be evidence that the quality of the generated texts is almost equivalent to that of the original.

7. Discussions

In Section 6, we found that the gap between M_{org} and M_{gen} was prominent. We also found that the scores for M_{gen} improved, and the gap between M_{org} and M_{gen} became less prominent with training data without multiple symptoms (labels). This indicates that the quality of the generated multi-symptom posts is lower than that of single-symptom posts.

7.1. Difficulty in Multi-symptom Handling

To find out what is the main cause of the gap between M_{org} and M_{gen}, we further analyzed the quality of generated text from different symptom combinations.

We analyze the qualitative difference of generated posts with single-symptom labels and multi-symptom labels. Table 6 shows examples of posts labeled "*Influenza and fever*", "*Hay fever and Runny nose*", and "*Cold and Runny nose*" from D_{gen} generated from the prompt SENTINEL in the upper half of the table. As a comparison to multi-symptom labels mentioned above, we provide examples of posts labeled with "*Fever*", "*Runny nose*", and "*Cold*" from D_{gen} generated from the prompt SENTINEL in the lower half of the table. Note that, **for this qualitative error analysis**, even if only the expression of "*Influenza*" is in the generated post, we consider the generated post correct for "*Influenza and Fever*". Similarly, for "*Hay fever and Runny nose*", we consider the generated post correct, even if only the expression of "*Hay fever*" is in the generated post. The reason is that such posts (only including expression of "*Influenza*" or "*Hay fever*" and labeled "*Influenza and Fever*" and "*Hay fever and Runny nose*") exist in D_{org}. We provide the correct examples in Table 6 (ids 1, 2, 6, 7, 11, and 12 for multi-symptom post generation and 16, 19, and 22

Model	Prompt	Accuracy	F1 (micro avg.)	Recall (micro avg.)	Precision (micro avg.)
$M_{gen(st)}$	SENTINEL	0.785 (0.0053)	0.837 (0.0111)	0.849 (0.0100)	0.825 (0.0041)
M_{org}	–	0.893 (0.0091)	0.932 (0.0122)	0.950 (0.0066)	0.915 (0.0067)

Table 5: Scores for $single_{only}$. Values in parentheses represent standard deviations of the scores from 10 models.

Multi-symptom	id (correct or error)	Generated post
Influenza and Fever	1 (correct)	高熱が出て、インフルエンザにかかってしまった。 (I got a high **fever** and caught **flu**)
	2 (correct)	昨日、インフルかかったから今日は休むわ。 (I'll take a break today because I've got a **flu** yesterday.)
	3 (error)	高熱で仕事できないから今日は休むわ。 (I can't work because of the high **fever**, so I'm taking a day off.)
	4 (error)	兄もインフルで寝込んでる。 (My brother is also in bed with the **flu**.)
	5 (error)	風邪ひいた。 (I've got a cold)
Hay fever and Runny nose	6 (correct)	花粉症で鼻水止まらない。 (I can't stop **running nose** with **hay fever**)
	7 (correct)	まさか自分が花粉症になるとは思ってもなかったわ。 (I never thought I would get **hay fever**.)
	8 (error)	鼻水止まらん。 (I can't stop my **runny nose**.)
	9 (error)	兄が花粉症で、今日も休むわ。 (My brother has **hay fever**, so I will be absent today.)
	10 (error)	そんなにひどい頭痛は久し振りだなあ。（I haven't had a headache that bad in a long time.)
Cold and Runny nose	11 (correct)	風邪で鼻水止まらない。 (I have a **cold** and can't stop my **runny nose**.)
	12 (correct)	鼻風邪ひいた。薬飲んだら眠くなってきた (I caught a **nose cold**. I fell asleep after taking medicine)
	13 (error)	風邪ひいたー。 (I caught a **cold**.)
	14 (error)	兄が鼻風邪で寝込んでる。 (My brother is in bed with a **nose cold**.)
	15 (error)	花粉症のせいか、鼻風邪が治らない。
Single-symptom	**id (correct or error)**	**Generated post**
Fever	16 (correct)	今年一番の熱。今日は仕事休むわ (The most horrible **fever** of the year. I'm off work today)
	17 (error)	弟が熱でて、仕事休むわ。 (My brother has a **fever**, so I will be absent from work.)
	18 (error)	これって風邪? (Is this a cold?)
Runny nose	19 (correct)	今日は鼻水止まらない。 (My **nose** won't stop **running** today.)
	20 (error)	兄が鼻水でぐったりしてる。 (My brother is exhausted from a **runny nose**.)
	21 (error)	日本には花粉症の人が多いんだってね。 (There are many people with hay fever in Japan.)
Cold	22 (correct)	また、風邪ひいたかも。 (I might have caught a **cold**, again.)
	23 (error)	中国で大流行した風邪が流行ってるらしいね。 (It seems that there is an epidemic of **cold** in China.)
	24 (error)	日本の夏は本当に寒いんだけど・・・・? (Summer in Japan is really cold, but...?)

Table 6: Examples of generated posts. The upper half is the examples of multi-symptom posts, and the lower half is examples of single-symptom posts

for single-symptom post generation).

We observed three types of typical errors.

Shortage error: The first type of error is a shortage of symptom expressions. In this type of error, even when given multiple conditions, such as "*Influenza and Fever*", generated posts only contain a part of symptom expressions. Examples are posts in ids 3, 8, and 13 in Table 6.

Out-of-user error: The second type of error is non-user symptom expressions. In this type of error, the posts are referring to a symptom of a non-user, rather than that of the X (previously Twitter) user who wrote the post. Examples are posts in ids 4, 9, and 14 in Table 6.

Label inconsistency error: The third type of error is those that include other symptoms. Examples are posts in ids 5, 10, and 15 in Table 6. Note that no symptom at all is also classified into this type of error.

We observed the same types of errors as multi-symptom post generation in single-symptom post generation. Since single-symptom post generation is supposed to satisfy only one condition, the first type of error observed in multi-symptom was not observed. Examples of the second type of error are shown in ids 17, 20, and 23 in Table 6, and the third type of error is shown in ids 18, 21, and 24.

7.2. Scores for Individual Symptom Combination

Since multi-symptom post generation has more complex conditions, more types of error can occur compared to single-symptom post generation. A possible reason for the lower scores for multi-symptoms is that the number of types of error in the multi-symptoms condition is larger than that in the single-symptom condition. We analyze the scores for individual symptom combinations.

The scores improved after the removal of multi-symptom labels. This suggests that the scores for multi-symptom labels are lower than those of single-symptom labels. Also, the difference in evaluation scores between $single_{all}$ and $single_{without}$ suggests that using generated multiple-symptoms posts in training had a negative influence even on classification of posts labeled with single and no symptom. We investigate those two assumptions by analyzing the scores for multi-symptom labels and single-symptom labels.

As mentioned in Section 3, multi-symptom la-

Model	Label	Accuracy	F1 (micro avg.)	Recall (micro avg.)	Precision (micro avg.)
$M_{gen(st)}$	*Influenza and Fever*	0.561 (0.0349)	0.701 (0.0055)	0.639 (0.0284)	0.776 (0.0218)
	Hay fever and Runny nose	0.874 (0.0126)	0.930 (0.0046)	0.912 (0.0111)	0.948 (0.0060)
	Cold and Runny nose	0.886 (0.0179)	0.952 (0.0020)	0.945 (0.0089)	0.960 (0.0047)
M_{org}	*Influenza and Fever*	0.754 (0.0346)	0.824 (0.0271)	0.779 (0.0325)	0.876 (0.0202)
	Hay fever and Runny nose	0.874 (0.0197)	0.904 (0.0171)	0.876 (0.0214)	0.934 (0.0122)
	Cold and Runny nose	**0.928** (0.0431)	**0.961** (0.0064)	0.956 (0.0307)	0.966 (0.0176)

Table 7: Metrics score for each multi-symptom label. Values in parentheses represent standard deviations of the scores from 10 models. The highest scores for accuracy and F1 are shown in bold.

		single-symptom labels				"*No symptom*"
Model	Training data	Accuracy	F1 (micro avg.)	Recall (micro avg.)	Precision (micro avg.)	Accuracy
$M_{gen(st)}$	MIX	0.900 (0.0208)	0.939 (0.0126)	0.931 (0.0137)	0.948 (0.0108)	0.562 (0.0161)
	SINGLE	0.907 (0.0179)	0.928 (0.0021)	0.907 (0.0179)	0.949 (0.0109)	0.723 (0.0161)
M_{org}	MIX	0.929 (0.0252)	0.953 (0.0148)	0.944 (0.0235)	0.963 (0.0133)	0.750 (0.0312)
	SINGLE	**0.950** (0.0207)	**0.962** (0.0160)	0.950 (0.0207)	0.974 (0.0110)	**0.802** (0.0245)

Table 8: Average metrics score for single-symptom labels and for "*No symptom*". Values in parentheses represent standard deviations of the scores from 10 models. The highest scores for accuracy and F1 are shown in bold.

bels with more than 30 posts are "*Cold and Runny nose*","*Hay fever and Runny nose*", and "*Influenza and Fever*". We present the scores for those three multi-symptom labels in Table 7. As shown in Table 7, only the multi-symptom combination "*Influenza and Fever*" has apparently different results between $M_{gen(st)}$ and M_{org}. This suggests that the other two combinations ("*Hay fever and Runny nose*" and "*Cold and Runny nose*") have less influence on the gap between $M_{gen(st)}$ and M_{org} in overall scores, and improvement after removal of posts with multi-symptom labels.

Scores for single-symptom labels, we found that most of them have similar results. The scores of M_{org} and $M_{gen(st)}$ from $single_{all}$ to $single_{without}$ tend to slightly increase compared to those of with multiple-symptom. We present the average scores of $M_{gen(st)}$ and M_{org} in $single_{all}$ and $single_{without}$ for "*Fever*", "*Runny nose*", "*Cold*", "*Diarrhea*", "*Headache*" and "*Cough*" in Table 8.

"MIX" represents that model is trained on mixed data of multi, single, and no symptom posts, and "SINGLE" represents that model is trained on only single and no symptom posts. As shown in Table 8, the scores of the four models do not differ much.

Only the label "*No symptom*" had different results from others. Due to this, we present the results for "*No symptom*" in Table 8. As shown in Table 8, the exact match accuracy for "*No symptom*" improved after the removal of posts with multi-symptom labels.

To summarize, the scores for multi-symptoms are lower than those of single-symptoms in general. Especially, the scores for the label "*Influenza and Fever*" was the lowest among three symptom combinations.

7.3. Prompting Methods in Post Generation

The results showed that adding the sentinel token to the prompts effectively improves the classification performance. The improvement implies that the quality of the data generated by Gen can be improved with proper instructions.

We explore the soft-prompting method in our experiment. Despite the findings in the previous work, we did not see an improvement from the baseline method. Although we did not analyze the reason for the underperformance of the soft prompting method, it would be interesting to investigate how we can apply the prompting methods usually used in natural language understanding tasks to generation tasks (such as experimenting with the different numbers of soft prompts). As mentioned in (Schick and Schütze, 2021), methods to avoid overfitting are necessary for prompting methods in future work.

8. Conclusions and Future Work

This study experimented with a method for generating a distributable surrogate corpus and investigated its viability. We experimented with different prompting methods in fine-tuning the pre-trained language generation model and evaluated the quality of generated corpora by the classification task. The results showed that when generating posts that contain multiple symptoms, the generated corpora suffer from the problem of semantic inconsistency between the labels and the generated content. Still, if the surrogate corpus was used in simpler settings, the generated data would be comparatively viable as a training corpus, as demonstrated in a single-symptom classification without using multi-symptom posts for training data.

In further pursuit of the research in this direction, we plan to 1) generate corpora in different languages than Japanese, namely *MedWeb*'s English and Chinese datasets, 2) measure the downstream performance in generating a larger amount of surrogate corpora than the original corpus, and 3) compare different models (other than T5) to investigate the impact of the choice of the model architecture on generation quality.

9. Limitations

Although we considered posts as a corpus in the medical domain, some clinical texts, such as discharge summaries, consist of much longer sentences. Since the pre-trained model used in this experiment accepts only less than 512 tokens, the low scalability to long texts, especially those with more than 512 tokens, is the limitation of this work.

10. Ethics Statement

The data used in this study, *MedWeb*, is deemed ethically sound. However, in the context of generating training data for medical NLP tasks, it is crucial to acknowledge the potential presence of errors in the generated data. Consequently, it is strongly advised against employing this data for tasks that have a direct impact on human life, such as automated diagnosis. Additionally, the study recognizes the possibility of the generated model memorizing and reproducing training data, emphasizing the importance of continuously integrating improvements based on relevant research findings.

11. Bibliographical References

Ali Amin-Nejad, Julia Ive, and Sumithra Velupillai. 2020. Exploring Transformer Text Generation for Medical Dataset Augmentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4699–4708.

Eiji Aramaki, Shoko Wakamiya, Shuntaro Yada, and Yuta Nakamura. 2022. Natural Language Processing: from Bedside to Everywhere. *Yearbook of Medical Informatics*.

Vincent Claveau, Antoine Chaffin, and Ewa Kijak. 2021. Generating artificial texts as substitution or complement of training data. *arXiv preprint*, arXiv:2110.13016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Udo Hahn and Michel Oleynik. 2020. Medical Information Extraction in the Age of Deep Learning. *Yearbook of Medical Informatics*, 29(1):208–220.

Julia Ive, Natalia Viani, Joyce Kam, Lucia Yin, Somain Verma, Stephen Puntis, Rudolf N Cardinal, Angus Roberts, Robert Stewart, and Sumithra Velupillai. 2020. Generation and evaluation of artificial mental health records for natural language processing. *NPJ digital medicine*, 3(1):69.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(160035):1–9.

Bosung Kim and Ndapa Nakashole. 2022. Data augmentation for rare symptoms in vaccine side-effect detection. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 310–315.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv preprint*, arXiv:1909.11942.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint*, arXiv:2107.13586.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT Understands, Too. *arXiv preprint*, arXiv:2103.10385.

Dimitris Pappas, Prodromos Malakasiotis, and Ion Androutsopoulos. 2022. Data Augmentation for Biomedical Factoid Question Answering. *arXiv preprint*, arXiv:2204.04711.

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.

Timo Schick and Hinrich Schütze. 2021. Few-Shot Text Generation with Natural Language Instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402.

Tawanda Sibanda and Ozlem Uzuner. 2006. Role of local context in automatic deidentification of ungrammatical, fragmented text. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 65–73.

Özlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the State-of-the-Art in Automatic De-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint*, arXiv:2201.05337.

Zheming Zuo, Matthew Watson, David Budgen, Robert Hall, Chris Kennelly, and Noura Al Moubayed. 2021. Data Anonymization for Pervasive Health Care: Systematic Literature Mapping Study. *JMIR Med Inform*, 9(10):e29871.

12. Language Resource References

Shoko Wakamiya, Mizuki Morita, Yoshinobu Kano, Tomoko Ohkuma, and Eiji Aramaki. 2019. Tweet classification toward twitter-based disease surveillance: New data, methods, and evaluations. *J Med Internet Res*, 21(2):e12783.