ProSwitch: Fine-Tuning Large Language Models to Generate Professional and Non-Professional Styled Text

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have been proven to be effective in various language tasks, such as text summarization and controlled text generation. However, research on the ability to switch between particular styles through fine-tuning LLMs is insufficient. In our study, we introduce an approach named **ProSwitch** to enable a language model to generate both professional and non-professional styled answers using knowledge-guided instruction tuning. ProSwitch is implemented in three stages: data preparation to gather domain knowledge and training set, instruction tuning to adjust language models with coarse and fine-grained instructions, and comprehensive evaluation to assess the professionalism discrimination and language quality of generated text. We compare the performance of ProSwitch with prevalent and specialized language models. The experimental results show that our approach achieves greater distinction between professional and non-professional text generation than the baseline models.

1 Introduction

Large Language Models (LLMs), such as ChatGPT and LLaMA (Touvron et al., 2023a), have excelled in natural language tasks including question answering (Omar et al., 2023; Tan et al., 2023; Baek et al., 2023) and information extraction (Perot et al., 2023; Schacht et al., 2023). Their proficiency in generating professional content for specialized scenarios allows them to integrate domain knowledge and deliver answers in a specific style, as seen with models like ChatDoctor (Li et al., 2023), ChatLaw (Cui et al., 2023) and FinGPT (Yang et al., 2023a). Importantly, an LLM should generate text in various styles, as shown in Figure 1, to satisfy both experts and laymen. An LLM-based question answering system that can discerningly produce both professional and non-professional content, according to context, can aid users in efficiently understanding and obtaining needed information.

Question:
Which cells express CIDEC protein in humans?

Professional Answer:
The cell death-inducing DNA fragmentation factor alpha-like effector c (CIDEC) is a lipid droplet-associated protein that promotes intracellular triglyceride (TAG) storage. CIDEC is highly expressed in adipocytes, but undetectable in normal liver. However, its hepatic expression rises during fasting or under genetic or diet-induced hepatosteatosis in patients.

Non-professional Answer:
CIDEC is essentially an expert in storing fats. It's all about fat cells. Yet, when you're fasting or indulging in lots of junk food, CIDEC surprisingly makes an appearance in the liver.

Figure 1: An example showing the answers in two styles with the same meaning for the same question, with terminologies highlighted.

The capability mentioned above refers to a specific aspect of controllable text generation, with the aim of customizing the text to suit various user needs (Pu and Demberg, 2023). In this task, a prompt outlining the desired style can be provided for a fine-tuned language model to produce content that closely imitates real scenarios. Despite the success of LLMs in numerous applications and the growing interest in controllable text generation (Hu and Li, 2021; Li et al., 2022; Pascual et al., 2021), there is a dearth of research exploring how LLMs can acquire style switching abilities. Furthermore, a quantitative evaluation is needed to measure the style discrimination of responses generated by LLMs. Therefore, our work investigates the following question: **Whether a proper fine-tuning procedure can improve an LLM's ability to switch between professional and non-professional styles, without compromising its foundational text generation capabilities**.

This study introduces **ProSwitch**, a method to improve the professional style switching capacity of an LLM through knowledge-guided tuning and evaluation. The process involves three stages, as depicted in Figure 2. We first collect text-based QA pairs from medical academic papers to form our positive dataset, characterized by its professional language style. We also gather domain-specific terminologies as knowledge for professional evaluation. Using GPT-4, we then enhance our training data by generating a mix of professional and non-professional pairs. In the instruction tuning phase, we create various prompts for a pretrained LLM to improve its style switching ability, ranging from coarse to fine-grained formulation. Fine-tuning parameters with these instructions helps the LLM distinguish between styles. We evaluate the adjusted LLM and baseline models using indicators that measure style switching ability and language quality. Our results indicate that our tuning method significantly improves style switching ability compared to prevalent and domain language models. The contributions of our research are as follows:

- We present **ProSwitch**, the first study on tuning LLMs to generate both professional and non-professional styles via LLM-augmented data preparation, multi-grained instruction tuning, and comprehensive evaluation.

- We propose indicators to evaluate professionalism discrimination and language quality of LLM-generated responses in a comprehensive evaluation.

- Our testing on medical QA datasets reveals that ProSwtich outperforms general and specialized LLMs switching professionalism styles without affecting fundamental generation capabilities.

2 Related Work

2.1 Text Style Transfer Learning

Text style transfer involves changing the style of an input sentence without altering its core meaning (Jin et al., 2022; Babakov et al., 2022; Mir et al., 2019). Previous studies have used sequence-to-sequence learning methods that apply parallel corpora with paired sentences in various styles (Cheng et al., 2020; Hu et al., 2021). However, due to the high demand for resources and costs for data labeling, parallel data in diverse styles is limited. This has encouraged a growing interest in investigating practical scenarios where only non-parallel stylized corpora are available (Reif et al., 2022; Malmi et al., 2020).

2.2 Controllable Text Generation

Controllable text generation is a rapidly developing field dedicated to creating text or responses with designated characteristics (Keskar et al., 2019; Dathathri et al., 2019; He et al., 2021). Various strategies have been suggested for this task, including sequence-to-sequence models that show potential in crafting excellent content tailored to particular needs. (Wu et al., 2021; Amplayo et al., 2021). Other methods have also been introduced to improve text generation controllability, such as conditional generation (He et al., 2021), prompt-based generation (Yang et al., 2023b), and multitask learning (Gu et al., 2022).

2.3 LLM Instruction Fine-Tuning

Instruction tuning is a straightforward technique that merges the attractive features of both the pretrain-finetune and prompting models through supervised fine-tuning. (Wei et al., 2021). Using the task-driven dataset, a pretrained model can be fine-tuned in a fully supervised way. The model is trained to sequentially predict each token in the output, given the instruction and input (Ouyang et al., 2022; Muennighoff et al., 2022; Taori et al., 2023; Berkeley et al., 2023). Meanwhile, some other domain language models apply instruction tuning methods to solve specific tasks or scenarios, such as information extraction (Wang et al., 2023), sentiment analysis (Varia et al., 2023), medical dialogue (Li et al., 2023), and code generation (Luo et al., 2023). To efficiently adapt LLMs to downstream tasks, efficient fine-tuning techniques optimize a small fraction of parameters in multiple ways, such as addition-based (Schick and Schütze, 2021), specification-based (Ben Zaken et al., 2022), and reparameterization-based (Hu et al., 2022).

Despite these progresses, the exploration of style-switching on professionalism of an LLM has not yet been addressed in existing studies. It remains to be seen whether a language model can produce text in both professional and casual styles through fine-tuning instructions with style-controlling prompts and domain knowledge.

3 Improving Style Professionalism Switching Skills

3.1 Task Formulation

We suppose to improve the ability of an LLM to switch between professional and non-professional styles, aiming to maximize the distinction between

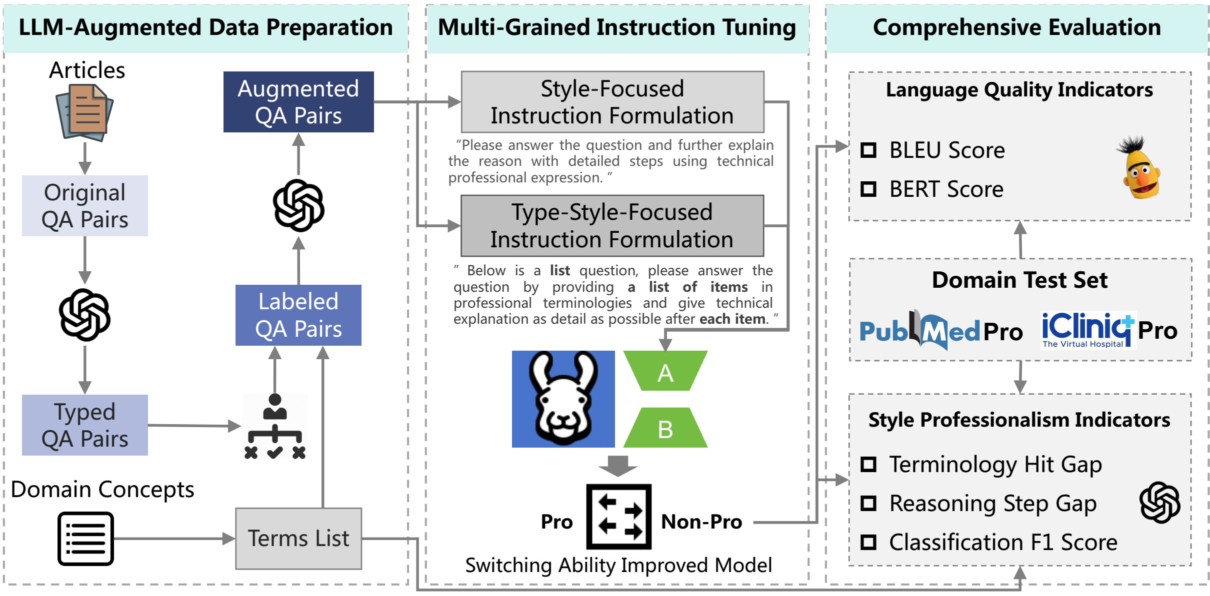

Figure 2: Our ProSwitch method contains three stages to improve the professionalism style switching ability of an LLM, through introducing domain data and knowledge for tuning and evaluation.

the text generated in two styles while maintaining the general language quality, by assessing with a set of detailed indicators.

Our objective can be formulated as:

$$\begin{aligned} m = \arg\max \big([P(O_p) - P(O_{np})] + \\ Q(O_p) + Q(O_{np}) \big), \\ O_p = LM(Pmt_p, I), \\ O_{np} = LM(Pmt_{np}, I) \end{aligned} \quad (1)$$

, where m is the desired method to maximize the score. $P(\cdot)$ and $Q(\cdot)$ are evaluation functions to calculate the professionalism and the quality of generated text, respectively. O_p and O_{np} are outputs generated by language model LM, which is provided with input I and prompts for professional style Pmt_p and non-professional style Pmt_{np}.

3.2 Prompt Formulation

A prompt to generate answers by an LLM in a particular style can be regarded as a concatenation of three components: task and style guidelines, the questions to be addressed, and the LLM-related limit information for output consistency. The prompt used in our study can be formulated as:

$$\begin{aligned} Pmt_p = Guide_p \parallel Q_n \parallel Limit_{lm}, \\ Pmt_{np} = Guide_{np} \parallel Q_n \parallel Limit_{lm} \end{aligned} \quad (2)$$

, where $Guide_p$ and $Guide_{np}$ are guidelines for generating professional and non-professional style answers. Q_n is the n-th question that need to be answered. $Limit_{lm}$ is the restrictive text for a specific language model lm. These components are connected with concatenation operators $\parallel$.

3.3 LLM-Augmented Data Preparation

3.3.1 Academic QA Pairs Collection

Text professional styles are often reflected in academic scenarios such as journal articles and conference papers, particularly in knowledge-intensive fields such as healthcare and medicine. Meanwhile, professional-style features can be learned from specialized QA tasks. With the information above, we collected two medical QA datasets, BioASQ (Tsatsaronis et al., 2015) and PubMedQA (Jin et al., 2019), sourced from academic articles. The responses in these datasets aim to clarify the questions based on a section of the related papers, which are rich in technical terms and detailed explanations. We consider these datasets as the seeds of our professional-style training data.

3.3.2 Question Type Classification

We have observed apparent style variations among different types of QA pairs. For instance, an answer using a list of terms to respond to a question differs significantly from an answer explaining a phenomenon. To help a model learn the unique features of diverse question categories, we categorize QA pairs by their question types. According to BioASQ, we consider four type of questions:

list, summarize, yesno, and factoid. However, PubMedQA does not specify question types, so we use GPT-4 to classify each QA pair into one of the four types, providing a few examples for reference. This LLM-supported type classification task can be formulated as:

$$T(Q_n) = LM(Pmt_t, (Q_n, A_n), L_t, \{S_1, ..., S_k\})$$
$$L_t = \{list, summarize, yesno, factoid\} \tag{3}$$

, where Q_n and A_n are the question and answer that need to be classified. Pmt_t is the instruction prompt to do the type classification task with type label set L_t. $S_1, ..., S_k$ is the set of examples for performing a few-shot learning, where k is the number of examples.

3.3.3 Data Balanced Augmentation

Due to the lack of corresponding non-professional or casual style responses in our dataset, and a shortage of QA pairs for training in both styles, we use GPT-4 for data augmentation using an in-context learning method (ICL) (Dong et al., 2022). Our goal is to increase the number of QA pairs for each question type in each style, striving for an adequate and equal size. GPT-4 is assigned to respond to questions using either professional or non-professional language, adhering to specific guidelines based on the presented questions and referring to provided examples. For professional data augmentation, GPT-4 is used exclusively to rephrase the referenced answers. In contrast, for non-professional data generation, GPT-4 directly provides an answer in casual language, complying with the provided guidelines. This data augmentation task can be formulated as:

$$\begin{aligned} A(Q_n) =& LM(Pmt_a, Q_n, \{S_1, ..., S_k\}), \\ Pmt_a =& f_i(Dict, L_p, T(Q_n)) \end{aligned} \tag{4}$$

, where Pmt_a is the instruction prompt for answering questions corresponding to question types and style labels. Pmt_a is retrieved from a pre-defined prompt dictionary $Dict$ by an indexing function f_i, using the type of the question $T(Q_n)$ and the professional label L_p as the keys.

With the aforementioned procedure, we produce both professional and non-professional QA pairs for each question type, ensuring equal size. This forms the training dataset of our method.

3.3.4 Concept Knowledge Processing

Unlike other style transfer learning studies, assessing the professionalism of an answer requires domain-specific expertise. In the medical field, we gather MeSH [1], a widely utilized XML-formatted list of medical terms. We derive all *QualifierNames* from the original file to compile a medical terminology list. This list is then used as our external domain knowledge to match terms in an answer for evaluating its professionalism.

3.4 Multi-Grained Instruction Tuning

3.4.1 Instruction Formulation

With the QA pairs generated in both professional and non-professional styles, we have to provide additional guidance to clarify the task for the language model during tuning. Adhering to the Alpaca (Taori et al., 2023) instruction format, we further create instructions with coarse and fine-grained descriptions for the style switching task. We present our instructions in two formats as following.

Style-focused (coarse-grained) instructions only consider the distinction between professional and non-professional styles. The instruction for professional answers is conveyed as: ***Answer the question and explain the reason with detailed steps using technical professional expressions.*** While for non-professional answers, the instruction is like: ***Answer the question and explain the reason with a simple explanation using casual non-professional expressions.*** By contrast, taking into account the significant distinction in responses to various question types, we further suggest a type-style-focused (fine-grained) instruction format by injecting type-based descriptions such as applying ***Answer the question with a list of items and explain each item...*** for the list questions. This formulation results in a permutation of two style labels (professional and non-professional) and four question types (list, summary, yesno, and factoid).

Different language models have varying capabilities and can generate texts in distinctive lengths and formats, leading to inconsistent comparisons. To address this problem, during our testing phase, we add brief restrictive information as described in Equation 2 to the input questions, guiding the language model to generate text in similar formats. Specifically, since some models fine-tuned with human chat data tend to provide lengthy responses, we append ***Answer the question directly with a single paragraph.*** to questions while inference to avoid unrelated information and dissimilar formats. For models fine-tuned with our prompts, we

[1] https://www.nlm.nih.gov/databases/download/mesh.html

include ***And why?*** to emphasize that more text of explanations is needed beyond the basic answer.

3.4.2 Instruction Tuning

With the prompts that contain the above instructions and restrictive information along with the input questions, we can fine-tune an open-source language model using various parameter optimizing methods. We apply LoRA (Hu et al., 2022), one of the parameter-efficient fine-tuning methods (PEFT) and full parameter fine-tuning (Radford and Narasimhan, 2018) in our task. LoRA significantly minimizes the number of trainable parameters by freezing the pre-trained model weights and incorporating trainable rank decomposition matrices into the Transformer layers. In contrast, full fine-tuning helps to maintain model quality and stability. In this study, we try both tuning methods to assess their effects on style-switching capabilities.

3.5 Comprehensive Evaluation

3.5.1 Professionalism Discrimination Scores

To evaluate the ability of ProSwitch in style switching, we propose a set of indicators to demonstrate the discrimination between professional and non-professional styles of the generated outputs.

The density of professional information, such as technical terms contained in a generated paragraph, is a useful metric that led us to introduce our first indicator, the **Terminology Hit Gap (THG)**. This measures the disparity between the number of technical terms found in professional and non-professional responses. As discussed in Section 3.3, we compute this indicator by matching the language model output with our medical domain concept list, noted as:

$$THG = |\frac{1}{N}\sum_{n=1}^{N} TH_n^p - \frac{1}{N}\sum_{n=1}^{N} TH_n^{np}|, \qquad (5)$$
$$TH_n^p = f_c(f_m(Term_d, LM(Pmt_n^p)))$$

, where TH_n^p and TH_n^{np} are the terminology hit values of the n-th answer in professional and non-professional styles, respectively. f_m and f_c are the functions for term matching and hit counting, respectively. $Term_d$ is the terminology list in domain d. $LM(Pmt_n^p)$ is the output generated by LM with a prompt describing the n-th question professionally.

Furthermore, we propose our second indicator to distinguish the level of reasoning of the generated language, called **Reasoning Step Gap (RSG)**, which measures the difference in the number of reasoning steps between professional and non-professional responses. This indicator is based on the notion that professional responses typically exhibit a more rigorous logical structure than informal language. To calculate RSG, we use GPT-4 to translate the raw text into sequential reasoning steps and then count these steps with a parsing function. This process can be noted below.

$$RSG = |\frac{1}{N}\sum_{n=1}^{N} RS_n^p - \frac{1}{N}\sum_{n=1}^{N} RS_n^{np}|, \qquad (6)$$
$$RS_n^p = f_p(LLM(Pmt_r, LM(Pmt_n^p)))$$

, where RS_n^p and RS_n^{np} are the reasoning step values of the n-th professional and non-professional answers. f_p is the parsing function to extract integer step counts from the reasoning details generated by an LLM. Pmt_r is the prompt for GPT-4 to perform the organization task from the give answer.

To further measure the ability of our fine-tuned language model to generate desired text styles, we implement a binary classification task to assess the performance of generated responses compared to their suggested professionalism labels with the commonly used F1 score, named as **Pro F1**.

3.5.2 Language Quality Scores

To investigate whether our tuning stage degenerates the fundamental ability of an LLM, we follow the metrics applied in (Sellam et al., 2020; Alihosseini et al., 2019) to measure the quality of the language generated with two indicators, including the BLEU score and the BERT score (Zhang et al., 2020), illustrated below.

$$BLEUscore =$$
$$min\left(1, \frac{Len(LM(Pmt_n))}{Len(Ref_n)}\right)\left(\prod_{i=0}^{m} P_i\right)^{\frac{1}{m}} \qquad (7)$$

, where Len is the function to calculate the length of text. $LM(Pmt_n)$ is the generated answer of the n-th question. Ref_n is the reference answers of the n-th question. P_i is the precision of the m-gram sequence that is taken into consideration while calculating the BLEU score.

$$BERTscore = 2\frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}},$$
$$P_{BERT} = \frac{1}{|\hat{x}|}\sum_{\hat{x_j}\in LM(Pmt_n)} \max_{x_i\in Ref_n} x_i^T \hat{x}_j, \qquad (8)$$
$$P_{BERT} = \frac{1}{|x|}\sum_{x_i\in Ref_n} \max_{\hat{x_j}\in LM(Pmt_n)} x_i^T \hat{x}_j,$$

where P_{BERT} and R_{BERT} are the precision and recall that calculated with the contextualized word embeddings for the reference answer and the generated output of the n-th question, respectively.

4 Experiment and Analysis

4.1 Dataset

We develop two domain datasets, **PubMedPro** and **IcliniqPro**, to assess the professional style switching ability. PubMedPro, which is constructed following the Alpaca format as detailed in Section 3.3, comprises 24,000 QA pairs in both professional and non-professional styles within the medical field. We select 40 questions in different types, with their corresponding answers in positive and negative styles, as our test set for evaluation. These questions originate from BioASQ (Tsatsaronis et al., 2015) and PubMedQA (Jin et al., 2019), two freely accessible question-answer datasets drawn from PubMed's academic articles. Another dataset is IcliniqPro, derived from icliniq[2], a medical dialogue dataset downloaded from the repositories mentioned in (Zeng et al., 2020; Wei et al., 2023). We manually and carefully select icliniq questions with the same number and similar expressions as PubMedPro, according to two principles: 1. The questions need to be answered with specific domain knowledge, and 2. The questions in direct expression without personal characteristics.

4.2 Baselines

We evaluate our ProSwitch method against three types of models. First, we use Llama2-Chat (Touvron et al., 2023b), a prevalent language model for general dialogue scenarios, as our competitor, which also serves as the foundation model of ProSwitch. Second, we compare with ChatDoctor (Li et al., 2023), a specialized language model fine-tuned with extensive patient-doctor dialogue data for improved accuracy of medical advice. Third, we assess ChatGPT [3], another language model for general purposes, but with much larger parameter size. We test two ProSwitch variants named ProSwitch-C and ProSwitch-F that represent our method with coarse and fine-grained instruction formulations, respectively.

[2]https://www.icliniq.com/qa/medical-conditions

[3]https://chat.openai.com/

4.3 Implementation Details

During the tuning phase, we train our ProSwitch model on 24,000 QA pairs, evenly distributed between two style labels and four question types. This training process, conducted on an NVIDIA RTX A6000 GPU for three epochs, has a learning rate of 2e-5 and a batch size of 128, taking roughly four hours in total. For evaluation, we establish a threshold for professionalism indicators by manually labeling 100 randomly selected QA pairs with two labels (professional or non-professional). The thresholds for terminology hit count and reasoning step count are then set at 2 and 4 respectively, aligning closely with human labeling results. We evaluate the ChatDoctor baseline with 13B parameters and Llama2-Chat with 7B parameters, identical to our ProSwitch. All experiments are conducted on the average of three times generation.

4.4 Professional Style Switching Performance

We assess ProSwitch and baselines using both professionalism discrimination and language quality metrics on the PubMedPro dataset. Additionally, we evaluate models on the IcliniqPro dataset, focusing solely on the professionalism discrimination indicators of the answers due to the absence of professional ground-truth answers to its questions. The experimental results are shown in Table 1. We can observe from the above results that:

ProSwith outperforms baselines on all datasets. Our knowledge-guided instruction tuning procedure improves the ability to switch between professional and non-professional styles, without compromising text generation capabilities.

Fine-grained instructions capture features better. The model that uses type-style-focused instructions provides more detailed guidance, which leads to the generation of text in styles with more expectations.

Specialized models downgrade the ability. ChatDoctor, though fine-tuned with domain dialogues, suffers a loss of style switching power on professionalism, which seems already equipped by Llama2 and ChatGPT.

Performance deficiency remains on large models. ChatGPT, though trained with enormous corpora and tuned with numerous tasks, still shows deficiency on the style switching task compared to our task-specific method.

Models	PubMedPro					IcliniqPro		
	Style Professionalism			Language Quality		Style Professionalism		
	THG	RSG	Pro F1	BLEU Score	BERT Score	THG	RSG	Pro F1
Llama2-Chat	2.92	0.58	0.63	0.2560	0.7292	2.28	0.62	0.51
ChatDoctor	1.74	0.33	0.60	0.2623	0.7204	1.68	0.89	0.44
ChatGPT	2.60	0.67	0.66	0.2964	0.7565	1.28	2.24	0.62
ProSwitch-C	3.44	0.74	0.70	**0.2998**	0.7472	3.38	3.04	0.76
ProSwitch-F	**4.04**	**1.06**	**0.73**	0.2955	**0.7676**	**3.58**	**3.31**	**0.81**

Table 1: The performance of two ProSwitch variants using coarse and fine-grained instructions, against Llama2, ChatDoctor, and ChatGPT on PubMedPro and IcliniqPro datasets. THG and RSG are our proposed professionalism discrimination indicators. Pro F1 is the F1 score of the stylistic binary classification. We only record professionalism indicators for IcliniqPro dataset as the absence of ground-truth professional answers as references. The optimal and suboptimal scores are highlighted with bold and underlined text, respectively.

4.5 Effect of Tuning Strategy

Apart from tuning language models using PEFT methods, we also attempt to train a ProSwitch model with full parameter fine-tuning approach, in order to investigate the potential capacity of a foundation language model to learn how to switch between professional and non-professional styles. The performance of ProSwitch using LoRA and full fine-tuning (FFT) methods is shown in Table 2.

Models	THG	RSG	Pro F1
ProSwitch-LoRA	3.04	1.06	0.73
ProSwitch-FFT	-1.10	1.35	0.76

Table 2: Professionalism discrimination indicators of ProSwitch trained with LoRA and full fine-tuning methods. Full fine-tuning tends to generate text with more reasoning steps but less terminology contained.

Surprisingly, the fully fine-tuned model tends to generate answers with more reasoning steps, but with fewer technical terminologies, leading to a very low THG score. This reveals that full fine-tuning learns expression better than wording.

4.6 Logical Density Analysis

Answers with more reasoning steps are regarded more professional, which can also lead to longer text. However, our statistical results in Table 3 demonstrate that the answers that contain many concise sentences linked logically can fit the professional style well. The density of reasoning within an answer is a notable feature of professionalism.

4.7 Case Study

In this section, we provide examples to highlight the improvement of our model after applying our tuning method. The examples in Table 4 showcases how ProSwitch and our base model, Llama2-Chat, differentiate technical terms and reasoning steps in both professional and non-professional styles. For a list question, our method is apt to generate more list items expressed in terminologies compared to the baseline model. For a yes/no question, ProSwitch tends to deliver more logically linked concise sentences than Llama2-Chat.

Models	Avg.Len	Avg.RS	RD
Llama2-Chat	418.5	5.29	0.013
ChatDoctor	443.5	5.83	0.013
ChatGPT	760.5	7.05	0.009
ProSwitch	336.0	5.92	**0.018**

Table 3: The average answer length (Avg.Len), average reasoning steps (Avg.RS), and reasoning density (RD) of the professional answers generated by different models. The higher reasoning density of ProSwitch indicates the more professional text it generates.

5 Conclusion

This study proposes ProSwitch, a method to improve the ability of language models to switch professionalism styles using knowledge-guided instruction tuning and evaluation. We focus on the medical domain and conducted three stages: LLM-augmented data preparation, multi-grained instruction tuning, and comprehensive evaluation to acquire and assess the professionalism discrimination and language quality of the generated responses. Our findings demonstrate that ProSwitch remarkably improves the style differentiation of generated text, compared to both prevalent and specialized language models.

Question	Professional Answer	Non-Professional Answer
Are there drugs for Tick-borne Encephalitis? (list)	**ProSwitch**: Yes, there are drugs for Tick-borne Encephalitis. These drugs include: 1. Ribavirin: This drug is used to ... 2. Rimantadine: This drug is used to ... 3. Oseltamivir: This drug is used to ... 4. Zanamivir: This drug is used to ... 5. Darunavir ...	**ProSwitch**: Yes, there are drugs for Tick-borne Encephalitis. These drugs are called antiviral medications and they help reduce the symptoms of the disease.
	Llama2-Chat: Yes, there are drugs for Tick-borne Encephalitis. The drugs used for treating TBE are doxycycline, ribavirin, and intravenous immunoglobulin. The choice of drug depends on the severity of the disease and the patient's age and medical history.	**Llama2-Chat**: Yes, there are drugs for Tick-borne Encephalitis. These drugs are called antiviral medications and they help reduce the symptoms of the disease.
Can Levetiracetam cause disturbance in sleeping? (yes/no)	**ProSwitch**: Yes, Levetiracetam can cause disturbance in sleeping.‖ This is due to the drug's ability to affect the activity of the neurotransmitter GABA,‖ which plays a crucial role in regulating sleep.‖ Levetiracetam can increase the activity of GABA,‖ leading to a decrease in the amount of time spent in slow-wave sleep and rapid eye movement (REM) sleep.‖ This can result in a decrease in the quality of sleep and an increase in daytime sleepiness.‖ Additionally, Levetiracetam can also cause vivid dreams and nightmares,‖ which can further disrupt ...	**ProSwitch**: Yes, Levetiracetam can cause disturbance in sleeping.‖ It can make you feel tired and sleepy during the day, ‖ but it can also make it harder to fall asleep at night.
	Llama2-Chat: Levetiracetam, a commonly used anticonvulsant medication, can potentially cause disturbances in sleep.‖ Here's why: ... ‖ While it is primarily used to treat epilepsy, it has also been shown to have anxiolytic and sedative effects in some individuals.‖ However, these effects can sometimes lead to disturbances in sleep patterns.	**Llama2-Chat**: Levetiracetam is a medicine that helps control seizures in people with epilepsy.‖ But, it can also make some people feel sleepy or have trouble sleeping.‖ This is because it affects the way the brain sends signals that help us fall asleep and stay asleep.‖ It's like when you're trying to ...

Table 4: Two examples to illustrate the discrimination of technical terms (highlighted with underlined text) and reasoning steps (separated by double vertical lines) between professional and non-professional answers generated by our ProSwitch model and Llama2-Chat model. Some content is omitted due to space constraints.

6 Limitations

The major limitation of our research is the lack of ground-truth QA pairs in a specific domain with professional styles. Future studies should be conducted to explore the transfer capabilities to different domains and the performance on larger foundation models.

7 Ethics Considerations

All datasets utilized in this study are publicly available and we have adhered to ethical considerations by not introducing additional information as input during LLM training and LLM text generation.

References

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Nikolay Babakov, David Dale, Varvara Logacheva, and Alexander Panchenko. 2022. A large-scale computational study of content preservation measures for text style transfer and paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

UC Berkeley, Stanford Cmu, and UC San. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90

Yu Cheng, Zhe Gan, Yizhe Zhang, Oussama Elachqar, Dianqi Li, and Jingjing Liu. 2020. Contextual text style transfer. *Cornell University - arXiv,Cornell University - arXiv*.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *International Conference on Learning Representations,International Conference on Learning Representations*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022. A distributional lens for multi-aspect controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1043, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junxian He, Wojciech Kryscinski, Bryan McCann, NazneenFatema Rajani, and Caiming Xiong. 2021. Ctrlsum: Towards generic controllable text summarization. *Cornell University - arXiv,Cornell University - arXiv*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiqiang Hu, Roy Ka-Wei Lee, and Charu C. Aggarwal. 2021. Syntax matters! syntax-controlled in text style transfer. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 566–575, Held Online. INCOMA Ltd.

Zhiting Hu and Li Erran Li. 2021. A causal lens for controllable text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 24941–24955. Curran Associates, Inc.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, page 155–205.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

NitishShirish Keskar, Bryan McCann, LavR. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv: Computation and Language,arXiv: Computation and Language*.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. *Cornell University - arXiv,Cornell University - arXiv*.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. *Cornell University - arXiv,Cornell University - arXiv*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, TevenLe Scao, MSaiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham-Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning.

Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Jiaqi Mu, Hao Zhang, and Nan Hua. 2023. Lmdx: Language model-based document information extraction and localization.

Dongqi Pu and Vera Demberg. 2023. ChatGPT vs human-authored text: Insights into controllable text summarization and sentence style transfer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 1–18, Toronto, Canada. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Sigurd Schacht, Sudarshan Kamath Barkur, and Carsten Lanquillon. 2023. Promptie - information extraction with prompt-engineering and large language models. In *HCI International 2023 Posters*, pages 507–514, Cham. Springer Nature Switzerland.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze questions for few shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artiéres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1).

Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Anna John, Rishita Anubhai, Smaranda Muresan, and Dan Roth. 2023. Instruction tuning for few-shot aspect-based sentiment analysis.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction.

Jason Wei, Maarten Bosma, VincentY. Zhao, Kelvin Guu, AdamsWei Yu, Brian Lester, Nan Du, AndrewM. Dai, and QuocV. Le. 2021. Finetuned language models are zero-shot learners. *Learning,Learning*.

Sibo Wei, Wenpeng Lu, Xueping Peng, Shoujin Wang, Yi-Fei Wang, and Weiyu Zhang. 2023. Medical question summarization with entity-driven contrastive learning.

Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision. *Cornell University - arXiv,Cornell University - arXiv*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023a. Fingpt: Open-source financial large language models.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2023b. Tailor: A soft-prompt-based approach to attribute-based controlled text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 410–427, Toronto, Canada. Association for Computational Linguistics.

Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, Hongchao Fang, Penghui Zhu, Shu Chen, and Pengtao Xie. 2020. MedDialog: Large-scale medical dialogue datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

A Prompts for Type Classification

We perform an LLM-based question type classification task by providing the following prompt 5 to GPT-4 and replacing the <question> variable with our real questions in our datasets.

You are tasked to classify a question into four types, following these guidelines: 1. Output the type of the question based on its form of asking. Possible types are: yesno, list, factoid, summary. 2. Just output one type without any descriptive information. Here are some examples: Question: Which DNA sequences are more prone for the formation of R-loops? Output: list Question: Are ultraconserved elements often transcribed? Output: yesno Question: What is clathrin? Output: summary Question: Which signaling pathway does sonidegib inhibit? Output: factoid Please output the type of the following question: Question: <question> Output:

Table 5: The prompt for question type classification.

B Prompts for Data Augmentation

We perform an LLM-based QA pair augmentation task by providing the following prompt 5 to GPT-4 and replacing <aim_style> with the style label we desire to collect. For a particular question type, we also provide the answer style description at the place of <answer_type> to restrict the generated text.

You are tasked to answer the question with <aim_style> language, following these guidelines: 1. You can refer to the provided examples to learn the differences between professional and non-professional answers. 2. You can refer to the original <style> answer and rephrase into a different <aim_style> answer. 3. For a <type> question, the <aim_style> answer usually <answer_style>. Here are examples of professional and non-professional answers: Question: What is gingipain? Professional answer: Porphyromonas gingivalis is a keystone periodontal pathogen that has been associated with autoimmune disorders. The cell surface proteases Lys-gingipain (Kgp) and Arg-gingipains (RgpA and RgpB) are major virulence factors, and their proteolytic activity is enhanced by small peptides such as glycylglycine (GlyGly). Question: Are reduced-nicotine cigarettes effective for smoking cessation? Non-professional answer: Yes, reduced-nicotine cigarettes are effective for smoking cessation. Please give a <aim_style> answer for the following question: Question: <question> Original <style> answer: <original_answer> Output:

Table 6: The prompt for QA pairs generation.

C Prompts for Reasoning Step Calculation

In our evaluation stage, we calculate the reasoning step count with the help of GPT-4 by reorganizing the answers into a step by step format and then giving the total step number at the end.

You are an assistant to explain the reasoning path of the answer. Here are some requirements: 1. Explain the reasoning path of the answer step by step with the content in both question and answer. 2. Provide the total steps at the last line, with the format: Total steps: <number>. Here is the question and the answer: Question: <question> Answer: <answer>

Table 7: The prompt for reasoning step reorganization.