ProSwitch: Knowledge-Guided Language Model Fine-Tuning to Generate Professional and Non-Professional Styled Text

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated efficacy in various linguistic applications, including text summarization and controlled text generation. However, studies into their capacity of switching between styles via fine-tuning remain underexplored. This study concentrates on textual professionalism and introduces a novel methodology, named **ProSwitch**, which equips a language model with the ability to produce both professional and non-professional responses through knowledge-guided instruction tuning. ProSwitch unfolds across three phases: data preparation for gathering domain knowledge and training corpus; instruction tuning for optimizing language models with multiple levels of instruction formats; and comprehensive evaluation for assessing the professionalism discrimination and reference-based quality of generated text. Comparative analysis of ProSwitch against both general and specialized language models reveals that our approach outperforms baselines in switching between professional and non-professional text generation.

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 content for specialized scenarios allows them to integrate domain knowledge and deliver answers in a specific style, as seen with models such as ChatDoctor (Li et al., 2023), ChatLaw (Cui et al., 2023) and FinGPT (Yang et al., 2023a). Meanwhile, an LLM should also generate text in various styles. As shown in Figure 1, a question-answering (QA) system that generates both professional and non-professional responses can help users access and grasp the needed information efficiently.

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.

However, there are few studies on characteristics of professional and non-professional language. Previous linguistic studies (Malyuga and Yermishina, 2021; Orrego-Carmona, 2016) state that non-professional or colloquial language uses analogies and phrasal verbs with fewer technological and organizational expressions. Another research in education (Malyuga, 2012) indicates that the distinctive feature of professional language is the terminological lexicon and the logical structure. The task of transferring between expert and layman text styles also aims to improve communication between scientific and general people, while existing studies (Pu and Demberg, 2023; Xu et al., 2022) focus only on lexical differences, neglecting sentence structure and domain knowledge.

Moreover, the technique for generating expert-layman text falls into controllable text generation, which focuses on tailoring text to meet diverse user requirements (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 controllable text generation (Hu and Li, 2021; Li et al., 2022; Pascual et al., 2021), there is a lack of research exploring how LLMs can acquire style switching abilities between professional and non-professional text. Furthermore, a quantitative evaluation is needed to measure the discrimination in style of the answers generated by LLMs. Therefore, our work investigates the following question: **Whether fine-tuning can improve an LLM's ability to switch between professional and non-professional styles, without compromising its text generation skills**.

This study introduces **ProSwitch**, a method to improve the professional style switching ability of an LLM through knowledge-guided tuning and evaluation. The process involves three stages, as shown in Figure 2. We first collect a dataset of text-based QA pairs from medical papers, featuring professional language, and gather domain-specific terms for professionalism evaluation. Using GPT-4 (OpenAI, 2023), we then enrich our training dataset with balanced professional and non-professional QA pairs. During instruction tuning, we craft prompts from multiple levels for a pretrained LLM to improve its style-switching ability. Based on previous studies, we perform a comprehensive evaluation of both professionalism discrimination and reference-based language quality of an LLM. Our findings indicate that ProSwitch can significantly improve the style-switching ability over existing general and domain LLMs.

In summary, our contributions are as follows: **(1)** We introduce **ProSwitch**, the first research on generating professional and non-professional text by exploiting external domain knowledge and internal knowledge from LLMs, different from the typical text style transfer tasks that concentrate only on lexical changes; **(2)** We propose and analyze our instruction-tuning strategy from multiple levels of instruction formatting for the task, which is distinctive from prompt-tuning and single-level instruction-tuning used in previous style transfer and text generation tasks; **(3)** We perform a comprehensive evaluation by proposing indicators from various aspects. Performance in QA datasets from the medical and IT domains reveals that ProSwtich outperforms general and specialized LLMs in the ability of switching between professional and non-professional text generation.

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 combines the best aspects of pretrain-finetune and prompting approaches via supervised fine-tuning. (Wei et al., 2021). In this way, a 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). 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 quickly adapt LLMs to downstream tasks, efficient fine-tuning techniques, such as addition-based (Schick and Schütze, 2021), specification-based (Ben Zaken et al., 2022), and reparameterization-based (Hu et al., 2022), optimize a small fraction of parameters.

Despite the advances described above, research has not explored the ability of LLMs to switch styles between professional and non-professional text guided by targeted prompts and domain-specific knowledge. This potential capacity of LLMs awaits further investigation.

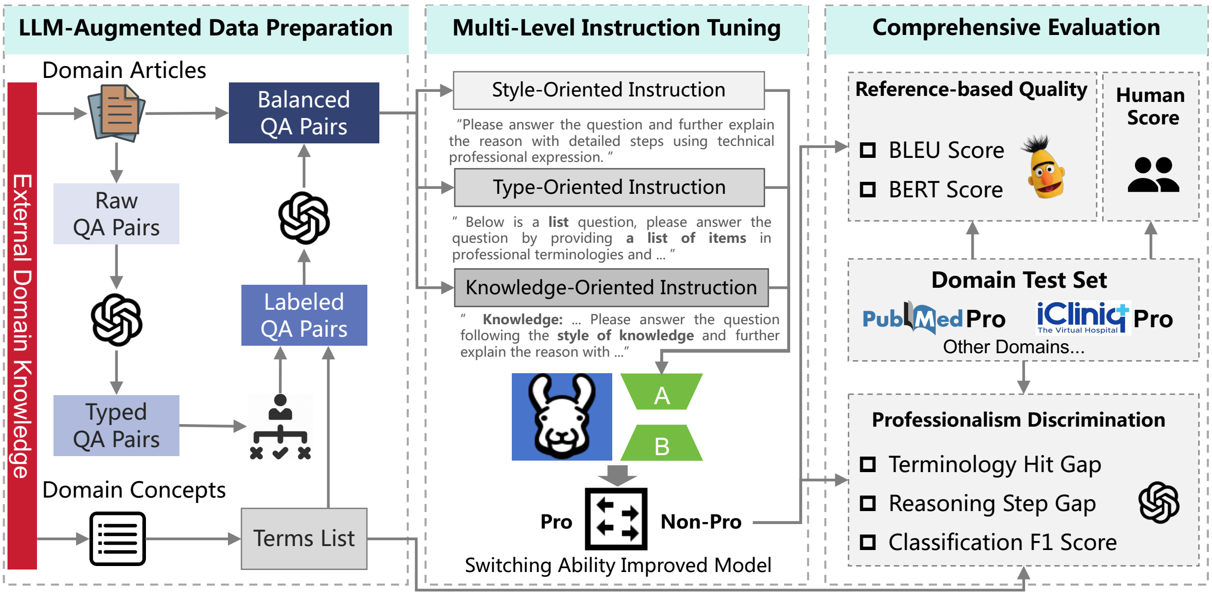

Figure 2: Our ProSwitch method contains three phases to improve the style switching ability in professionalism, through exploiting domain knowledge for instruction tuning in multiple levels and performance evaluation.

3 Preliminaries

3.1 Professionalism Definition

Referring to previous studies on linguistics and education (Malyuga and Yermishina, 2021; Orrego-Carmona, 2016; Malyuga, 2012), the professionalism of a sentence should consider two aspects of features, including terminology and logical structure. These two features can be quantified by counting the number of domain terms and the reasoning steps. Then, a sentence can be classified as a professional answer if these two metrics reach a combined threshold, denoted as:

$$Pro(O) = \begin{cases} 1 & \text{if } f_t(O, L_{\mathcal{T}}) \geq a \wedge f_r(O) \geq b, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $f_t(\cdot)$ and $f_r(\cdot)$ are functions to calculate terms and reasoning steps, respectively, from the output sentence O. $L_{\mathcal{T}}$ is the list of terms to be matched. a and b are threshold values.

3.2 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 the text generated in two styles while maintaining the quality of generated sentences, by assessing with a set of detailed indicators. Our objective can be formulated as:

$$\begin{aligned} &\max \left(f_p(O_p, O_{np}) + f_q(O_p) + f_q(O_{np}) \right), \\ &O_p = LM(Pmt_p), O_{np} = LM(Pmt_{np}), \end{aligned} \tag{2}$$

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

3.3 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, questions to be addressed, and LLM-related limit information for output consistency. The prompt used in our study can be formulated as:

$$\begin{aligned} Pmt_p &= Guide_p \parallel Q_n \parallel Limit_{lm}, \\ Pmt_{np} &= Guide_{np} \parallel Q_n \parallel Limit_{lm} \end{aligned} \tag{3}$$

, 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 the concatenation operator $\parallel$.

4 Proposed ProSwitch

4.1 LLM-Augmented Data Preparation

Academic QA Pairs Collection. Textual 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 related papers, which are rich in technical terms and detailed explanations. We consider these datasets as the seeds of our professional-style training data.

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. This inspires us to categorize QA pairs by their question types to help a model learn the type-related features of professionalism. According to BioASQ, we consider four types: list, summarize, yes/no, and factoid. However, PubMedQA does not specify any types, so we employ GPT-4 to classify each QA pair into one of the four types by providing a few demonstrations, followed by a manual check. This LLM-supported type classification task can be formulated as:

$$\begin{gathered} T(Q_n) = LM(Pmt_t, (Q_n, A_n), L_t, \{S_1, ..., S_k\}) \\ L_t = \{list, summarize, yes/no, factoid\} \end{gathered} \tag{4}$$

, 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.

Data Balanced Augmentation. Due to the lack of corresponding non-professional style responses in our dataset and a shortage of QA pairs for training in both styles, we are urged to perform data augmentation for the following training phase. Using LLM and in-context learning (ICL) (Dong et al., 2022), our goal is to increasingly generate 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 follows:

$$\begin{aligned} A(Q_n) =& LM(Pmt_a, Q_n, \{S_1, ..., S_k\}), \\ Pmt_a =& f_i(Dict, L_p, T(Q_n)) \end{aligned} \tag{5}$$

, 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.

Term Knowledge Processing. Unlike other style transfer learning studies, evaluating the professionalism of an answer in our task requires domain-specific expertise, and terms in a domain help us perform the evaluation automatically. 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 phrases in an answer to quantitatively evaluate its professionalism.

4.2 Instruction Formulation

4.2.1 Multi-Level Instruction

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 focusing on three levels of information for the style-switching task, presented as follows.

Style-oriented instruction. First, we only apply the superficial description of 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.* For non-professional answers, the instruction is: *Answer the question and explain the reason with a simple explanation using casual non-professional expressions.*

Type-oriented instruction. In contrast, taking into account the significant differences in responses

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

to various question types, we suggest a type-oriented instruction format by providing type-based descriptions such as applying *Answer the question with a list of items and explain each item with...* for the list-type questions. This formulation results in a permutation of two style labels (professional and non-professional) and four question types (list, summary, yes/no, and factoid).

Knowledge-oriented instruction. Furthermore, with the rich expression information contained in domain-related articles, we propose a knowledge-oriented instruction by injecting article snippets (implicit professional style knowledge) related to the question to construct professional instructions, formatted as: *Knowledge: <article_snippet>. Answer the question following the style of the knowledge provided and ...*. For nonprofessional instructions, we inject a more descriptive sentence as explicit non-professional style knowledge to explain what the answer should be expressed, formatted as: *Knowledge: A non-professional answer is prone to use analogies and phrasal verbs to explain the question with fewer technological and organizational expressions. Answer the question following the knowledge using non-professional expressions.*

4.2.2 LLM-Related Restrictive Information

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

4.3 Comprehensive Evaluation

4.3.1 Professionalism Discrimination Scores

To evaluate the ability of ProSwitch, we propose a set of indicators to demonstrate the discrimination between professional and non-professional styles of the generated output. Referring to the professionalism defined in Section 3.1, we describe our indicators as follows.

Terminology Hit Gap (THG). The number of technical terms contained in a generated paragraph is a useful metric that leads us to introduce our first indicator. THG measures the disparity between the number of technical terms found in professional and non-professional responses. With the term knowledge collected, we compute this indicator by performing a phrase-level matching between the output of the language model and phrases in our domain term list, noted as:

$$\begin{aligned} THG &= |\frac{1}{N}\sum_{n=1}^{N} TH_n^p - \frac{1}{N}\sum_{n=1}^{N} TH_n^{np}|, \\ TH_n^p &= f_c(f_m(Term_d, LM(Pmt_n^p))) \end{aligned} \tag{6}$$

, 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.

Reasoning Step Gap (RSG). Furthermore, we propose our second indicator to distinguish the level of reasoning of the generated language, RSG, which measures the number gap 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 casual language. To calculate RSG, we use GPT-4 to transform the raw answer into sequential reasoning steps and then count these steps with a parsing function. This process can be noted below.

$$\begin{aligned} RSG =& |\frac{1}{N}\sum_{n=1}^{N} RS_n^p - \frac{1}{N}\sum_{n=1}^{N} RS_n^{np}|, \\ RS_n^p =& f_p(LLM(Pmt_r, LM(Pmt_n^p))) \end{aligned} \tag{7}$$

, 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.

Pro F1. 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 actual professionalism labels with the commonly used F1 score (Forman et al., 2003), denoted as Pro F1.

4.3.2 Reference-based 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 as follows:

$$BLEUscore = \\ min\left(1, \frac{Len(LM(Pmt_n))}{Len(Ref_n)}\right)\left(\prod_{i=0}^{m} P_i\right)^{\frac{1}{m}} \quad (8)$$

, 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.

$$BERTscore = 2\frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}}, \quad (9)$$

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.

5 Experiment and Analysis

5.1 Dataset

We develop two domain datasets, **PubMedPro** and **IcliniqPro**, to assess the style switching ability. PubMedPro, which is constructed following the Alpaca format as detailed in Section 4.1, 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 QA datasets drawn from PubMed's academic articles[2]. Another dataset is IcliniqPro, derived from iCliniq[3], a medical dialogue dataset downloaded from the repositories mentioned in (Zeng et al., 2020; Wei et al., 2023). We manually and carefully select questions with the same number and similar expressions as those in PubMedPro, according to two principles: 1. The questions need to be answered with specific knowledge; 2. The questions are stated directly without personal feelings.

5.2 Baselines

We evaluate ProSwitch variants against multiple baselines. Llama2-Chat (Touvron et al., 2023b), our foundation model, is a prevalent language model for general dialogue scenarios. ChatDoctor (Li et al., 2023) is a specialized language model fine-tuned with extensive patient-doctor dialogue data to improve the accuracy of medical advice. ChatGPT 3.5 [4] is another language model for general purposes, but with a much larger parameter size. Furthermore, we deploy an implementation in reference to the work on formality style transfer (FST) (Etinger and Black, 2019) as a traditional baseline. We reproduce the expert-layman work (Pu and Demberg, 2023) as a competitor in controllable text summarization (CTS).

5.3 Instruction Tuning

With the prompts that contain multiple levels of instructions and restrictive information, as described in Section 4.2, we apply LoRA (Hu et al., 2022), one of the parameter-efficient fine-tuning methods (PEFT), to optimize a pretrained model for our task. LoRA significantly minimizes the number of trainable parameters by freezing the pretrained model weights and incorporating trainable rank decomposition matrices into the Transformer. Finally, using Llama2-Chat as our base model, three optimized models are generated according to the three levels of instruction formulations, named ProSwitch-S, ProSwitch-T, and ProSwitch-K, respectively.

5.4 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, the thresholds for terminology hit count and reasoning step count are set at 1 and 4 respectively, aligning closely with human labeling results with AUC greater than 0.85. We evaluate the ChatDoctor baseline with 13B parameters and Llama2-Chat with 7B parameters, identical to our ProSwitch. All experiments

[2] https://pubmed.ncbi.nlm.nih.gov/

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

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

are conducted on average three times generation. More details of the implementation are described in Appendix B.

5.5 Professional Style Switching Performance

We evaluate ProSwitch and the baselines on the PubMedPro dataset using both professionalism discrimination and reference-based quality metrics. Meanwhile, we evaluate models on the IcliniqPro dataset, focusing solely on professionalism discrimination 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 performance 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-oriented and knowledge-oriented instructions can provide more detailed guidance, leading to the generation of text in styles with higher expectations.

Specialized models downgrade the ability. ChatDoctor and FST, though fine-tuned with domain data or targeted at similar tasks, suffer a loss of style switching on professionalism, which appears already partially equipped by other baselines.

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

5.6 Human Evaluation

We recruit student volunteers to perform the evaluation as a crowdsourcing task. Each question and its generated answers in the test set are rated in two aspects: style discrimination and language fluency. For style discrimination, each pair of answers is scored 1–5 points to determine the degree to which the two answers can be distinguished in professionalism. For language fluency, each answer is also rated 1–5 for how it can be understood grammatically. Following (Xu et al., 2022), we calculate the percentage of ratings with 4 and 5 points as success rates (SR), and also the average scores (AS) of each criterion. Human evaluation is performed on all baseline models and ProSwitch variants. The results of the human evaluation shown in Table 2 demonstrate consistency with our indicator-based evaluation, in general.

Models	**Discrimination**		**Fluency**	
	AS	**SR**	**AS**	**SR**
Llama2-Chat	3.60	0.57	3.97	0.78
ChatDoctor	3.02	0.38	<u>4.27</u>	**1.00**
ChatGPT	3.52	0.55	**4.40**	**1.00**
FST	2.38	0.08	3.32	0.51
CTS	3.18	0.47	4.11	<u>0.97</u>
ProSwitch-S	3.91	0.78	4.05	0.92
ProSwitch-T	**4.28**	<u>0.90</u>	4.13	**1.00**
ProSwitch-K	<u>4.23</u>	**0.93**	4.25	**1.00**

Table 2: Human evaluation results. AS and SR represent the average score and the success rate, respectively. Optimal and suboptimal scores are highlighted with bold and underlined text, respectively. ProSwitch models are confirmed to generate answers with more professionalism discrimination and maintain language fluency.

5.7 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-T using LoRA and full fine-tuning (FFT) methods is shown in Table 3. Surprisingly, the fully fine-tuned model tends to generate answers with more reasoning steps, but with far fewer technical terms, leading to a very low THG score. We discover that the FFT model tends to generate long logical sentences but with plain words as professional answers, indicating that fully fine-tuning learns expression better than wording.

Models	**THG**	**RSG**	**Pro F1**
ProSwitch-T (LoRA)	4.04	1.06	0.73
ProSwitch-T (FFT)	-1.10	1.35	0.76

Table 3: 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 far fewer terms.

5.8 Logical Density Analysis

As professional language is generally treated as a logically structured system (Malyuga, 2012), answers with more reasoning text are considered

Models	PubMedPro					IcliniqPro		
	Style Professionalism			Reference-based 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
FST	0.48	0.46	0.62	0.1859	0.6948	-	-	-
CTS	1.68	0.84	0.65	0.2732	0.7322	1.32	2.57	0.69
ProSwitch-S	3.44	0.74	0.70	0.2998	0.7472	3.38	3.04	0.76
ProSwitch-T	**4.04**	1.06	0.73	0.2955	0.7676	**3.58**	3.31	**0.81**
ProSwitch-K	3.26	**2.32**	**0.77**	**0.3349**	**0.7799**	3.30	**3.84**	0.79

Table 1: The performance of three ProSwitch variants using three levels of instruction formulations, against LLMs, style transfer model (FST), and controllable text summarization method (CTS) on two 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.

more professional, which can also lead to longer text. However, our statistical results in Table 4 show that answers that contain many logically linked concise sentences can fit the professional style well. By contrast, the density of reasoning steps within an answer is an effective feature of professionalism.

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 4: 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.

5.9 Domain Adaptation Analysis

We evaluate the adaptation performance of ProSwitch in another domain: IT technical support, using the TechQA dataset (Castelli et al., 2020), which contains real-world questions posed by users on the IBMDeveloper forum. This domain offers multifaceted professional support to accommodate users with diverse needs. We randomly select a subset of TechQA IT questions as the test set and collect 352 IT terms and their synonyms from various websites to calculate the THG indicator. We directly employ the ProSwitch models that were trained with medical QA datasets to generate professional and non-professional answers for IT questions. The performance compared to general models such as Llama2-Chat and ChatGPT is shown in Table 5.

Models	Pro F1	THG	RSG
Llama2-Chat	0.46	0.18	0.83
ChatGPT	0.42	0.15	1.38
ProSwitch-S	0.57	0.24	2.85
ProSwitch-T	0.63	0.29	3.02
ProSwitch-K	**0.69**	**0.36**	**4.07**

Table 5: The performance of ProSwitch variants on answering technical questions from IT domain without training with domain datasets. Our method consistently generates answers with more professional and non-professional features.

6 Conclusion

This study proposes ProSwitch, a knowledge-guided instruction tuning method, to improve the ability of language models to switch between professional and non-professional text generation. We focus on domain QA tasks and conduct three phases: LLM-augmented data preparation, multi-level instruction tuning, and comprehensive evaluation to acquire the ability of professionalism discrimination and reference-based quality. Our findings demonstrate that ProSwitch remarkably improves the style differentiation of generated text, compared to both general and specialized baselines.

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 more different domains and the performance on larger foundation models.

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

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. The TechQA dataset. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1269–1278, Online. Association for Computational Linguistics.

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.

Isak Czeresnia Etinger and Alan W Black. 2019. Formality style transfer for noisy, user-generated conversations: Extracting labeled, parallel data from unlabeled corpora. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pages 11–16.

George Forman et al. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res., 3(Mar):1289–1305.

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.

Elena N Malyuga. 2012. Professional language in formal and business style. Global Journal of Human Social Science, 12(3):7–10.

Elena N Malyuga and Valentina E Yermishina. 2021. The expressive function of colloquialisms in professional discourse: The linguopragmatic aspect. In E3S Web of Conferences, volume 284. EDP Sciences.

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.

OpenAI. 2023. Gpt-4 technical report.

David Orrego-Carmona. 2016. A reception study on non-professional subtitling: Do audiences notice any difference? Across Languages and Cultures, 17(2):163–181.

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.

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. Fine-tuned 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.

Wenda Xu, Michael Saxon, Misha Sra, and William Yang Wang. 2022. Self-supervised knowledge assimilation for expert-layman text style transfer. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 11566–11574.

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 LLM Prompts

A.1 Prompts for Type Classification

We perform an LLM-based question type classification task by providing the following prompt in Table 6 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. 3. Summary questions are usually more general, but factoid questions are more specific. 4. You can infer the type according to the display forms of possible answers. 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 6: The prompt for question type classification.

A.2 Prompts for Data Augmentation

We perform an LLM-based QA pair augmentation task by providing the following prompt in Table 7 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 7: The prompt for QA pairs generation.

A.3 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. The reorganization prompt is shown in Table 8.

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 8: The prompt for reasoning step reorganization.

B More Implementation Details

B.1 LLM-Augmented Type Classification

The type of question is classified into one of the four categories by GPT-4. However, there are some problems while performing the process. The major problem is the confusion between the meaning of types. One confusion occurs between factoid and summary, as they have similar sentence structures, such as the beginning of "What is". The difference is that summary questions are usually more general, such as "What is Synucleinopathy?" and "What is a zoonotic virus?". However, factoid questions are more specific and aim to obtain a particular aspect of an entity, such as "What is the function of a viral peplomer?". Another confusion is between the list and factoid questions, which also have similar expressions but have different formats of answers. To address the above problems, as shown in Appendix A.1, we provide guidelines in our instruction to describe the distinctive information to help GPT4 better understand the differences.

B.2 LLM-Augmented Data Balancing

As our PubMedPro dataset is constructed from academic QA scenarios, there are far more professional QA pairs than non-professional ones. To balance the number of QA pairs in each style, we perform an LLM-augmented data generation using the prompt shown in Table 7. What needs to be emphasized is that we have tried different types of style description to guide GPT4 to generate or rephrase into our desired answers. One type of description follows our style-oriented instruction format, such as *explains the reason with detailed steps using technical professional expressions* for a professional question. Another type of description follows our type-oriented instruction format, such as *has a list of items and explains each item with reasons in detailed steps using technical professional expressions* for a professional and list-type question. We evaluate the output of these two types of prompt for data generation and rephrasing by random sampling and manual checking, and select the type-oriented description as the final version for data augmentation, since it facilitates generation closer to reference answers.

B.3 Indicator Threshold Setting

We search for the thresholds of professional indicators according to human-labeled samples for the construction of training datasets. We recruit volunteers to manually label a small part of randomly selected QA pairs and screen out consistent labeling answers with three labels (professional, non-professional, unsure). We then adjust the number of terminology count and reasoning steps of these answers that can distinguish professional and non-professional answers in order to fit the labels generated by humans. A distribution visualization of the two indicators is shown in Figure 3, demonstrating a larger number of reasoning steps than the terminology count should be specified to identify professional responses.

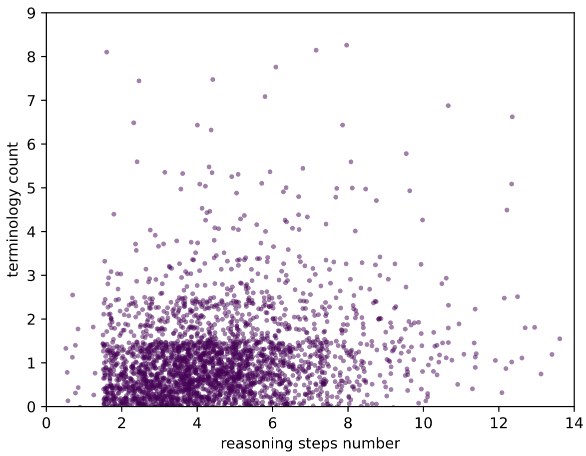

Figure 3: Distribution of terminology count and reasoning step count from a part of PubMedPro dataset. Each value is added with a small random number for visual differentiation.

B.4 Human Evaluation Details

We recruit volunteers for human evaluation of our generated text from postgraduate students of our university. We provide the instruction as follows: *You are asked to rate answers generated by an LLM for a text generation research in domain QA scenario. Please evaluate the English answers provided in terms of their professional distinctiveness and language fluency as follows: 1. For the given pair of English answers, evaluate their professional differentiation; the better the distinction in professionalism, the higher the score (1-5 points). 2. For each English answer provided, rate the fluency of the language; the easier it is to read, the higher the score (1-5 points).* We keep the rating results that are completed by the annotators and calculate the average scores and success rates for each model, as discussed in the experiment section.

B.5 Result Variation Caused by LLM

There are two reasons that can cause unwanted performance variations in our task. The first comes from the formulation of prompts, which affects generating professional and non-professional answers the most. To avoid model-specific generation bias, we remove all predefined system prompts in open source language models while fine-tuning and inference. To avoid competitors from generating too short or too long answers, we test extensive examples and add restrictive information related to the model. Another reason that may lead to variational results is from multiple requests, which affects both answer generation and reasoning step decomposition. We run each model three times and calculate the average values of the indicators or the number of steps as usable results. We also set appropriate hyperparameters such as Top-K, Top-P, and Temperature for each model during generation.

C Case Study

In this section, we provide examples to highlight the improvement of our model after applying our tuning method. The examples in Table 9 showcases how ProSwitch and our base model Llama2-Chat generate answers to a question with a different number of technical terms. ProSwitch is apt to generate more domain terms compared to the baseline model when providing a professional response. Table 10 shows that ProSwitch tends to deliver more logically linked concise sentences than Llama2-Chat while answering a yes/no question in professional style. Table 11 shows that ProSwitch can generate an answer with more domain terms and a better sentence structure in the meantime than Llama2-Chat.

Question	Professional Answer	Non-Professional Answer
What is the biological role of Neddylation?	**ProSwitch**: The biological role of Neddylation is to regulate the activity of enzymes involved in ... This process involves the covalent attachment of a small protein called NEDD8 to the target enzymes ... The NEDD8 protein is a member of the ubiquitin-like protein family and ... The Neddylation pathway is a highly regulated process that plays a crucial role in maintaining cellular homeostasis and ...	**ProSwitch**: Neddylation is when proteins get a special thing called a Nedd8 tag. This tag helps the proteins do their job and also helps them get broken down. It's important for the cell to know when to break down proteins and when to keep them around.
	Llama2-Chat: Neddylation is a post-translational modification that plays a crucial role in various cellular processes, including protein function, localization, and degradation. It's a way for proteins to be modified after they're made, which can affect how they work and how long they last. The biological role of Neddylation can be broken down into several key functions ...	**Llama2-Chat**: Neddylation is like a molecular glue that helps proteins do their thing in the body. It adds a little tag to proteins that helps them interact with other proteins and stuff. Like, it's like a secret handshake for proteins!

Table 9: An example to illustrate the differentiation of technical terms (highlighted with underlines) in professional and non-professional answers generated by our ProSwitch model and Llama2-Chat model. Some content is omitted due to space constraints.

<table>
<tr><th>Question</th><th>Professional Answer</th><th>Non-Professional Answer</th></tr>
<tr><td rowspan="2">Can Levetiracetam cause disturbance in sleeping?</td><td>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 ...</td><td>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.</td></tr>
<tr><td>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.</td><td>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 ...</td></tr>
</table>

Table 10: An example to illustrate the differentiation of reasoning steps (separated by double vertical lines) in professional and non-professional answers generated by our ProSwitch model and Llama2-Chat model. Some content is omitted due to space constraints.

Question	Professional Answer	Non-Professional Answer
Are there transposon-free regions in mammalian genomes?	**ProSwitch**: Yes, there are transposon-free regions in mammalian genomes. These regions are known as transposon-free regions (TFRs) and are found in both the human and mouse genomes. Despite the presence of over 3 million transposons ... The majority of human TFRs are found to be similar to their orthologous counterparts in the mouse, indicating that ... Additionally, many human TFRs also overlap with orthologous TFRs in the marsupial opossum, suggesting that these regions have been conserved across different mammalian lineages ...	**ProSwitch**: Yeah, there are some parts of mammalian genomes that don't have transposons. These regions are called "transposon-free regions" or "TFRs". They're like special zones in the genome where transposons don't hang out.
	Llama2-Chat: Yes, there are transposon-free regions in mammalian genomes. These regions are known as "gap regions" or "gap-like sequences" and are found in the vicinity of transposon insertion sites. To explain further, transposons are mobile genetic elements that can insert themselves into new locations within a genome. When a transposon inserts itself into a region of the genome that is already occupied by a gene or regulatory element, it can disrupt the function of that gene or element ...	**Llama2-Chat**: Yeah, there are some regions in mammalian genomes that are free of transposons. These regions are usually found near the centromere or at the ends of chromosomes.

Table 11: An example to illustrate the differentiation of technical terms (highlighted with underlines) and conjunctive adverbs (highlighted with wave underlines) for logical structure in professional and non-professional answers generated by our ProSwitch model and Llama2-Chat model. Some content is omitted due to space constraints.