SRCB at #SMM4H 2024: Making Full Use of LLM-based Data Augmentation in Adverse Drug Event Extraction and Normalization

Hongyu Li[1], Yuming Zhang[1], Yongwei Zhang[1], Shanshan Jiang[1], and Bin Dong[1]

[1]Ricoh Software Research Center (Beijing) Co., Ltd
{Hongyu.Li, Yuming.Zhang1, Yongwei.Zhang,
Shanshan Jiang, Bin Dong}@cn.ricoh.com

Abstract

This paper reports on the performance of SRCB's system in the Social Media Mining for Health (#SMM4H) 2024 Shared Task 1: extraction and normalization of adverse drug events (ADEs) in English tweets. We develop a system composed of an ADE extraction module and an ADE normalization module which further includes a retrieval module and a filtering module. To alleviate the data imbalance and other issues introduced by the dataset, we employ 4 data augmentation techniques based on Large Language Models (LLMs) across both modules. Our best submission achieves an F1 score of 53.6 (49.4 on the unseen subset) on the ADE normalization task and an F1 score of 52.1 on ADE extraction task.

1 Introduction

The Social Media Mining for Health (#SMM4H) Workshop has served as a competitive platform aimed at promoting the development and evaluation of advanced natural language processing (NLP) systems for the detection, extraction and normalization of health related information in social media texts (tweets, reviews and Reddit posts). Among the shared tasks in #SMM4H 2024 (Xu et al., 2024), ADE extraction and normalization has been the longest running task, which requires participants first extract spans of adverse drug events (ADEs) expressions from tweets and then normalize the spans to MedDRA[1] ontology's preferred terms (PTs). This task is evaluated in the following order of priority: the F1 score of ADE normalization, the F1 score of ADE normalization on the unseen subset[2] and F1 score of ADE extraction.

The challenges of this task lie in: (1) The dataset exhibits extreme imbalance between samples containing ADEs (positive) and those not containing ADEs (negative). (2) The validation and test sets contain ADEs to which the corresponding PTs are not seen during training. (3) The given texts are tweets with frequent use of informal grammar as well as irregular vocabulary.

In recent years, LLMs have been widely used for data augmentation (Cai et al., 2023; Whitehouse et al., 2023; Zhang et al., 2024) to improve data quality, especially for the unbalanced and noisy data. To this end, we propose to make full use of LLM-based data augmentation using GLM-4[3] and GPT-3.5 across both ADE extraction and ADE normalization modules. For ADE extraction, we first enrich the training set from both mention-level and context-level by prompting the LLMs to rewrite ADE spans and the other parts of the positive samples. To cover more ADEs with unseen preferred terms, we prompt LLMs to generate synthetic tweets and obtain pseudo ADE annotations by our previously trained ADE extraction models. For ADE normalization, we ask the LLMs to rewrite the given tweets into formal written texts. In addition, we also ask the LLMs to give an explanation for each lowest level term (LLT) and preferred term (PT) in MedDRA dictionary. These LLM-based data augmentation techniques show varying degrees of performance improvement on our system. Finally, our system obtained the highest ADE normalization F1 score in task 1.

2 System Description

we employ a relatively comprehensive pipeline for data pre-processing detailed in Appendix A. Our system includes an ADE extraction module which extracts spans of ADEs from the given tweets and an ADE normalization module which normalizes the extracted spans to PTs. We further decompose ADE normalization into two steps, namely MedDRA term (LLTs, PTs) retrieval and MedDRA

[1] https://www.meddra.org/

[2] The ADEs to which the corresponding preferred terms are not seen during training.

[3] https://open.bigmodel.cn/

Proceedings of The 9th Social Media Mining for Health Applications (#SMM4H) Workshop and Shared Tasks, pages 32–37
August 15, 2024 ©2024 Association for Computational Linguistics

term filtering. MedDRA term retrieval involves retrieving up to 20 MedDRA terms by conducting similarity search bewteen the embeddings of each extracted span and the LLTs and PTs. Given the input tweet, an extracted span and each of the retrieved LLTs or PTs, a MedDRA term filtering model is asked to classify whether the retrieved LLT or PT happens as an ADE in the tweet or not. The base models we use are detailed in Appendix B.

3 Data Augmentation Using LLMs

Since the dataset poses challenges such as data unbalance, unseen PTs and domain mismatch as illustrated before, we utilize 4 different LLM-based (GLM-4, GPT-3.5) data augmentation techniques across ADE extraction and ADE normalization. More details are provided in D

3.1 ADE Mention Rewriting and Context Rewriting

To address the issue of unbalanced data, we increase the number of positive samples by leveraging LLMs to rewrite ADE mentions and their contexts separately, and then combining them. The process includes: (1) ADE Mention Rewriting: We prompt GLM-4 to rewrite each ADE mention while keeping the remaining parts unchanged three times. This gives us four versions of each ADE mention (the original plus three new ones). (2) Context Rewriting: We mask the ADE mentions in the tweets and prompt GLM-4 to rewrite the remaining parts, creating three new versions of each tweet. This results in four different tweet templates for each original tweet (the original plus three new ones). (3) Combining Rewritten Mentions and Contexts: We sample up to three combinations[4] of ADE mentions and insert each combination into a randomly chosen rewritten tweet template, generating new positive samples. By using 3 different random seeds during the sampling process, we create three distinct augmented training sets, leading to more model candidates trained with different datasets during ensemble.

3.2 LLM Synthetic Tweets with Pseudo ADE Annotations

We note that there are ADEs of unseen PTs in the validation and test sets. Therefore we prompt the LLMs to generate synthetic tweets with more ADE expressions which may be corresponding to unseen PTs. We first extract all potential drugs in the train and validation sets using GLM-4 and then validate whether the extracted drugs are genuine drugs with GPT-3.5 to get rid of the noise. Meanwhile, we request GPT-3.5 to list the common side effects of each validated drug. We then prompt GPT-3.5 to generate 3 tweets based on a given drug and one or two sampled side effects of it. We sample the side effects for 3 times, thus obtain 9 tweets for each drug ideally. Finally, we get pseudo ADE annotations by conducting majority voting over the predictions of 27 ADE extraction models, which are trained with data augmented by ADE mention rewriting and context rewriting[5].

3.3 Tweet rewriting

In the provided tweets, informal grammar and irregular vocabulary are frequently used. This typically results in performance degradation in language models pre-trained with a general-domain corpus such as BERT (Devlin et al., 2019). Therefore, we prompt GLM-4 to rewrite the tweets into formal written texts easier to understand. The rewritten tweets are used during MedDRA filtering.

3.4 MedDRA Term Explanation

Considering explanation of the MedDRA terms (LLTs, PTs) may benefit context understanding in MedDRA filtering, we prompt GPT-3.5 to give an explanation for each MedDRA term. During MedDRA term filtering, we append the description of the given MedDRA term to the end of the input sequence.

4 Results

4.1 Experiment Results

ADE extraction The evaluation results on the validation set of our models are illustrated in Table 1. The ADE extraction F1 scores shown in the table are averaged over models trained with random seeds of 42, 21 and 1, and also averaged over all models trained with 3 different MR-CR augmented training sets with different sampling seeds. A significant performance improvement is observed for both techniques of LLM-based data augmentation. And the scores are further improved

[4] We did not utilize all combinations, as doing so would bias the data distribution towards samples with a higher number of entities.

[5] 27 models: using 3 different augmented training sets, fine-tuned based on RoBERTa-large, BERTweet-large and DeBERTa-v3-large with random seeds of 42, 21 and 1

Traing data	PLM	ADE Extraction F1 (Dev)
ORIG[†]	RoBERTa-large	67.16
	BERTweet-large	67.51
	DeBERTa-v3-large	71.97
ORIG+MR-CR	RoBERTa-large	70.82
	Bertweet-large	71.16
	DeBERTa-v3-large	74.43
ORIG+SynTweet	RoBERTa-large	71.93
	BERTweet-large	72.88
	DeBERTa-v3-large	73.07
ORIG+MR-CR +SynTweet	Roberta-large	72.10
	BERTweet-large	72.01
	DeBERTa-v3-large	75.39

Table 1: The ADE extraction F1 scores on validation set for our ADE extraction models. ORIG represents the original training set. † represents the baseline models. MR-CR denotes the augmented data created by ADE Mention Rewriting and Context Rewriting. SynTweet denotes the LLM-generated tweets.

through combining both techniques on RoBERTa-large and DeBERTa-v3-large. Among the 3 pre-trained language models used, DeBERTa-v3-large outperforms the other two by a considerable margin.

ADE normalization We select the ensemble ADE extraction results with highest F1 score as the input for ADE normalization to retrieve MedDRA terms. The MedDRA term retrieval model using fine-tuned text embedding model achieves a recall of 85.88, higher than 78.82 without fine-tuning. The evaluation results of our MedDRA term filtering models are illustrated in Table 2. By introducing Tweet Rewriting and MedDRA Term Explanation, the performance is improved in most of our experiments. Furthermore, the models using only rewritten tweets achieved a more impressive improvement than the ones that include original tweets in their inputs. This indicates that the informal grammar as well as irregular vocabulary in the original tweets hinder model learning, and converting the original tweets into formal written texts can effectively address this issue.

4.2 Test Results

Table 3 shows the submission results of our system. Among the submission files, **submission-2** uses the ensemble ADE extraction result over all models trained from DeBERTa-v3-large except for the baseline models. We observe a smaller proportion of positive samples in the ensemble ADE extraction result on the test set, compared to the training and validation sets. Therefore, we add more ADE predictions for **submission-1** and **submission-3** by including the ensemble result of a smaller number of model candidates (9 models based on DeBERTa-v3-large w/ ORIG+MR-CR+SynTweet). In addition, **submission-1** and **submission-2** are ensembled with MedDRA filtering models achieving an F1 score over 77. For **submission-3**, we use the model combination that obtains the highest F1 score on validation set out of all possible combinations.

Input sequence	PLM	ADE Normalization F1 (Dev)
ORIG[†]	RoBERTa-large	74.73
	BERTweet-large	73.63
	DeBERTa-v3-large	75.82
ORIG&TE	RoBERTa-large	75.14
	Bertweet-large	74.03
	DeBERTa-v3-large	76.24
ORIG&TR	RoBERTa-large	74.87
	BERTweet-large	73.91
	DeBERTa-v3-large	76.50
ORIG&TR&TE	Roberta-large	77.53
	BERTweet-large	76.24
	DeBERTa-v3-large	77.35
TR&TE	Roberta-large	78.89
	BERTweet-large	77.53
	DeBERTa-v3-large	79.33

Table 2: The ADE normalization F1 scores on validation set for our ADE normalization models. The first column compares the differences in input sequences other than the extracted spans and retrieved MedDRA terms. ORIG represents using the original training tweets. † represents the baseline models. & represents the concatenation operation of sequences. TR denotes the Tweet Rewriting sequence. TE denotes the Term Explanation sequence.

Submission	ADE Normalization F1	ADE Normalization F1 (Unseen)	ADE Extraction F1
1	53.6	49.4	52.1
2	52.7	48.9	51.8
3	53.3	49.1	52.1
Official Baseline	43.9	32.3	48.1
Mean	28.3	20.9	32.7
Median	29.3	14.1	37.6

Table 3: Submission results on test set

5 Conclusion

In this work, we propose to use 4 different LLM-based data augmentation techniques for Task 1, including ADE mention rewriting and context rewriting, synthetic tweets with psuedo annotations, tweet rewriting and MedDRA term explanation. As a result, our system, including an ADE extraction module and an ADE normalization module, achieves the highest F1 score (53.6) in Task 1 among all the teams. For future work, modification on the models will be studied to achieve more model-level improvements. Additionally, pipelines based solely on LLMs will be further explored.

References

Alham Fikri Aji, Made Nindyatama Nityasya, Haryo Akbarianto Wibowo, Radityo Eko Prasojo, and Tirana Fatyanosa. 2021. Bert goes brrr: a venture towards the lesser error in classifying medical self-reporters on twitter. In *Proceedings of the Sixth Social Media Mining for Health (# SMM4H) Workshop and Shared Task*, pages 58–64.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Xunxin Cai, Meng Xiao, Zhiyuan Ning, and Yuanchun Zhou. 2023. Resolving the imbalance issue in hierarchical disciplinary topic inference via llm-based data augmentation. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1424–1429. IEEE.

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

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Xi Liu, Han Zhou, and Chang Su. 2022. Pingantech at smm4h task1: Multiple pre-trained model approaches for adverse drug reactions. In *Proceedings of The Seventh Workshop on Social Media Mining for Health Applications, Workshop & Shared Task*, pages 4–6.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Andrey Sakhovskiy, Zulfat Miftahutdinov, and Elena Tutubalina. 2021. Kfu nlp team at smm4h 2021 tasks: Cross-lingual and cross-modal bert-based models for adverse drug effects. In *Proceedings of the Sixth Social Media Mining for Health (# SMM4H) Workshop and Shared Task*, pages 39–43.

Darius Koenig Julius Lipp Sean Lee, Aamir Shakir. 2024. Open source strikes bread - new fluffy embeddings model.

Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. 2023. Llm-powered data augmentation for enhanced cross-lingual performance. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Dongfang Xu, Guillermo Lopez Garcia, Lisa Raithel, Rolland Roller, Philippe Thomas, Eiji Aramaki, Shuntaro Yada, Pierre Zweigenbaum, Sai Tharuni Samineni, Karen O'Connor, Yao Ge, Sudeshna Das, Abeed Sarker, Ari Klein, Lucia Schmidt, Vishakha Sharma, Raul Rodriguez-Esteban, Juan Banda, Ivan Flores Amaro, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2024. Overview of the 9th social media mining for health applications (#SMM4H) shared tasks at ACL 2024. In *Proceedings of The 9th Social Media Mining for Health Research and Applications Workshop and Shared Tasks*, Bangkok, Thailand. Association for Computational Linguistics.

Meishan Zhang, Gongyao Jiang, Shuang Liu, Jing Chen, and Min Zhang. 2024. Llm–assisted data augmentation for chinese dialogue–level dependency parsing. *Computational Linguistics*, pages 1–24.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61.

A Data Pre-Processing

We follow similar data pre-processing steps with (Aji et al., 2021; Sakhovskiy et al., 2021; Liu et al., 2022). We utilize Ekphrasis[6] (Baziotis et al., 2017) package as well as some customized processing steps. In addition to lowercase, (1) we remove the "@USER", "HTTPURL" and the following placeholders. (2) We remove emojis, but convert the emoticons into natural language (e.g. ":)" to "<smile>"). (3) We convert the slangs into natural language (e.g. "lol" to "laugh out loud"). (4) We replace the HTML entities into corresponding symbols (e.g. "&amp;" to "&" and "&lt;" to "<"). We then convert all ampersand symbols "&" into "and". (5) We conduct hashtag unpacking, which performs word segmentation on the content following a "#" symbol. (6) We employ spell correction.

B Base Models

B.1 ADE Extraction Models

The ADE extraction task can be treated as a Named Entity Recognition (NER) task with only one entity label, namely "ADE". Therefore, we implemented a span-based NER model following prior works (Lee et al., 2017; Luan et al., 2018, 2019; Zhong and Chen, 2021). We use a pre-trained language model (PLM: RoBERTa, BERTweet or DeBERTa) as encoder to obtain contextualized representation $\mathbf{X}_t$ for each input token $x_t \in X$. Consequently, the span representation $\mathbf{h}_e(s_i)$ for each potential span $s_i \in S$ is denoted as:

$$\mathbf{h}_e(s_i) = \left[\mathbf{X}_{START(i)}; \mathbf{X}_{END(i)}; \phi(s_i)\right],$$

where $\phi(s_i) \in \mathbb{R}^{d_F}$ denotes the length embeddings of s_i. The span representation $\mathbf{h}_e(s_i)$ is then fed into a softmax layer to predict the probability distribution of "the span is an ADE" and "the span is not an ADE".

B.2 ADE Normalization Models

MedDRA term retrieval We implement a basic dense retrieval model with LangChain[7] and Faiss[8] (Douze et al., 2024; Johnson et al., 2019) libraries for efficient similarity search of dense vectors. We use a fine-tuned mxbai-embed-large-v1[9] (Sean Lee, 2024; Li and Li, 2023) to obtain text embeddings for both queries (extracted spans) and MedDRA terms. Details of text embedding model fine-tuning could be found at Appendix C. We decide to retrieve up to 20 LLTs and PTs, which strike a balance between the performance of retrieval model and the subsequent filtering model.

MedDRA term filtering We simply utilize a PLM-based (RoBERTa, BERTweet or DeBERTa) binary classification model to classify whether each retrieved MedDRA term happens as an ADE in the given tweet. The basic inputs of our models is the concatenation of the given tweet, a extracted span as an potential ADE and one of the retrieved MedDRA terms. We collect the training samples for MedDRA term filtering models by using the MedDRA term retrieval model to retrieve 20 MedDRA terms for each gold-annotated ADE mention. We add the gold MedDRA term to the retrieved ones if it is not within them. Besides, we collect more negative samples by adding the ADE predictions from our earlier 5-fold cross-validation experiments.

C Text Embedding Model Fine-tuning

We use the example training script for NLI tasks provided by Sentence-Transformers (Reimers and Gurevych, 2019) to fine-tune mxbai-embed-large-v1. To construct the fine-tuning data, we first group the LLTs and PTs with the same PTID (ID of a preferred term). We obtain the "entailment" samples by pairing each gold ADE span in the training set with each of the LLTs and PTs belonging to the same group with its gold LLT or PT. Then for each "entailment" sample, we construct a "contradiction" sample by pairing the gold ADE span with a PT or LLT belonging to another randomly sampled group.

D Details of LLM-based Data Augmentation

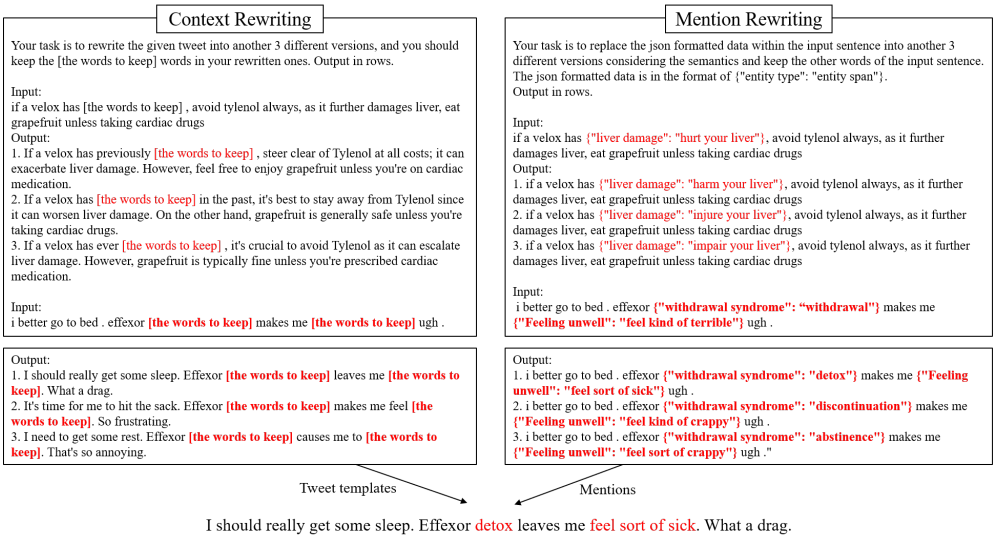
Figure 4 shows an example of LLM-based data augmentation using mention rewriting and context rewriting. For ADE mention rewriting, we first convert the ADE mentions in the given tweets into JSON-formatted data indicating the gold LLTs or PTs (e.g. "withdrawal" to {"withdrawal syndrome": "withdrawal"}). We then prompt GLM-4 to output the entire input tweet while only replacing the ADE mentions in the JSON-formatted data for 3 times. These measures ensure that GLM-4 does not change the PT corresponding to each ADE

[6] https://github.com/cbaziotis/ekphrasis

[7] https://github.com/langchain-ai/langchain

[8] https://github.com/facebookresearch/faiss

[9] https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1



Your task is to rewrite the given tweet into another 3 different versions, and you should keep the [the words to keep] words in your rewritten ones. Output in rows.

Input:
if a velox has [the words to keep] , avoid tylenol always, as it further damages liver, eat grapefruit unless taking cardiac drugs
Output:
1. If a velox has previously [the words to keep] , steer clear of Tylenol at all costs; it can exacerbate liver damage. However, feel free to enjoy grapefruit unless you're on cardiac medication.
2. If a velox has [the words to keep] in the past, it's best to stay away from Tylenol since it can worsen liver damage. On the other hand, grapefruit is generally safe unless you're taking cardiac drugs.
3. If a velox has ever [the words to keep] , it's crucial to avoid Tylenol as it can escalate liver damage. However, grapefruit is typically fine unless you're prescribed cardiac medication.

Input:
i better go to bed . effexor **[the words to keep]** makes me **[the words to keep]** ugh .

Output:
1. I should really get some sleep. Effexor **[the words to keep]** leaves me **[the words to keep]**. What a drag.
2. It's time for me to hit the sack. Effexor **[the words to keep]** makes me feel **[the words to keep]**. So frustrating.
3. I need to get some rest. Effexor **[the words to keep]** causes me to **[the words to keep]**. That's so annoying.

Your task is to replace the json formatted data within the input sentence into another 3 different versions considering the semantics and keep the other words of the input sentence. The json formatted data is in the format of {"entity type": "entity span"}.
Output in rows.

Input:
if a velox has {"liver damage": "hurt your liver"}, avoid tylenol always, as it further damages liver, eat grapefruit unless taking cardiac drugs
Output:
1. if a velox has {"liver damage": "harm your liver"}, avoid tylenol always, as it further damages liver, eat grapefruit unless taking cardiac drugs
2. if a velox has {"liver damage": "injure your liver"}, avoid tylenol always, as it further damages liver, eat grapefruit unless taking cardiac drugs
3. if a velox has {"liver damage": "impair your liver"}, avoid tylenol always, as it further damages liver, eat grapefruit unless taking cardiac drugs

Input:
i better go to bed . effexor **{"withdrawal syndrome": “withdrawal"}** makes me **{"Feeling unwell": "feel kind of terrible"}** ugh .

Output:
1. i better go to bed . effexor **{"withdrawal syndrome": "detox"}** makes me **{"Feeling unwell": "feel sort of sick"}** ugh .
2. i better go to bed . effexor **{"withdrawal syndrome": "discontinuation"}** makes me **{"Feeling unwell": "feel kind of crappy"}** ugh .
3. i better go to bed . effexor **{"withdrawal syndrome": "abstinence"}** makes me **{"Feeling unwell": "feel sort of crappy"}** ugh ."

I should really get some sleep. Effexor detox leaves me feel sort of sick. What a drag.

Figure 1: An example of LLM-based data augmentation using mention rewriting and context rewriting

Assuming you are a Twitter user, your task is to write at least 3 tweets with less than 30 words complaining the side effects of “**[side_effects]**" of “**[drug_name]**" that happens to you.
The expressions for the side effects should be colloquial and varied, and you are also recommended to use words other than the original terms.

Examples:
1. I found the humira to fix all my crohn's issues, but cause other issues. i went off it due to issues w nerves/muscle spasms
2. ciprofloxacin: how do you expect to sleep when your stomach is a cement mixer?
3. debating on taking a trazodone and literally passing out for the day.
4. well played tysabri...kicking butt #nosleep.

Output:

Figure 2: Prompt template for LLM-based data augmentation using LLM synthetic tweets

Your task is to rewrite the input tweet for better flow within 30 words. You should make it easier to understand and you should not use any abbreviations.

Input: if # avelox has hurt your liver , avoid tylenol always , as it further damages liver , eat grapefruit unless taking cardiac drugs.
Output: If Avelox has caused liver damage, always avoid Tylenol as it can further harm the liver. Also, avoid grapefruit if taking cardiac drugs.

Input: apparently , baclofen greatly exacerbates the " " ad " " part of my attention deficit hyperactivity disorder . average length of focus today : about 30 seconds .
Output: Baclofen makes my attention deficit hyperactivity disorder worse. I can only focus for around 30 seconds.

Input: **[tweet]**
Output:

Figure 3: Prompt template for LLM-based data augmentation using tweet rewriting

Your task is to explain the given medical term in one sentence using simple and understandable vocabulary.

Input: Abnormal weight gain
Output: Abnormal weight gain refers to gaining an unusually large amount of weight, which may be a sign of health issues or imbalances in the body.

Input: **[LLT or PT]**
Output:

Figure 4: Prompt template for LLM-based data augmentation using MedDRA term explanation

mention and maintains the coherence of texts after ADE mention rewriting. For context rewriting, we use "[the words to keep]" to mask the ADE mentions and ask GLM-4 to rewrite the other parts of the given tweets into another 3 versions. In this way, GLM-4 will pay more attention to the part-of-speech of the masked words while rewriting the contexts, thus ensuring the smoothness of the rewritten contexts when inserting the rewritten ADE mentions. Prompt templates used in synthetic tweets, tweet rewriting and MedDRA term explanation are illustrated in Figure 2,3 and 4.

E LLM-based ADE Extraction

We abandoned LLM-based ADE Extraction for the following reasons: (1) The extracted ADEs are often absent from the original texts and include a large number of false positives, making it impossible to ensemble with other models. (2) Despite filtering the LLM extraction results with other models', the retrieval module achieves a lower recall of 82.35 compared to 85.88 achieved on the ensemble results of PLM-based ADE extraction models.

F Model Ensemble

We employ majority voting for both ADE extraction and ADE normalization. For ADE extraction, we set the threshold at $\frac{1}{3}$ of the number of model candidates. For ADE normalization, we collect all LLTs or PTs classified into "happens as an ADE" by at least one model candidate for each extracted span. We then map the LLTs to PTs and the span is labeled with the specific PT that has the highest counts.