Leveraging Large Language Models for Biomedical Terminology Normalization

Anonymous ACL submission

Abstract

Biomedical Terminology Normalization aims at finding the standard term in a given termbase for non-standardized mentions coming from social media or clinical texts, and the mainstream approaches adopted with the "Recall and Rerank" framework. Instead of the traditional pretraining-finetuning paradigm, we would like to explore the possibility of accomplishing this task through a training-free paradigm using the powerful large language models (LLMs). Hoping to address the costs of re-training due to discrepancies of both standard termbases and annotation protocols. Another major obstacle in this task is that both mentions and terms are short texts. Short texts contain an insufficient amount of information that can introduce ambiguity, especially in a biomedical context. Therefore, besides using the advanced embedding model, we distill knowledge from LLM to expand the short text for a more informative description, enabling a superior unsupervised retrieval approach. Furthermore, we introduce an innovative training-free biomedical terminology normalization framework. By leveraging the reasoning capabilities of the LLM, in combination with supervised data and domain-specific expertise, to conduct more sophisticated ranking and re-ranking processes. Experimental results across multiple datasets indicate that both our unsupervised and supervised approaches achieve state-of-the-art.

1 Introduction

Biomedical Terminology Normalization is a basic research task in clinical natural language processing, linking non-standard mentions extracted from social media or clinical texts to normalized terms in a standard termbase, e.g., UMLS, MedDRA, ICD, SNOMED CT, to find the standard terms that have the same semantics as them. (Ruch et al., 2008; Leaman et al., 2013; Leal et al., 2015; Luo et al., 2019; Lee and Uzuner, 2020).

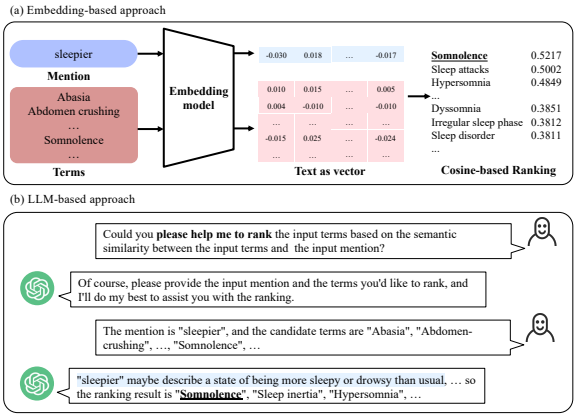

Figure 1: An Example of Embedding-based Approach and Probing Large Language Models for Terminology Normalization Tasks.

Mainstream approaches tend to adopt the "recall and rerank" framework to accomplish this task, i.e., recall some candidates from the full standard database first and then re-rank them more finely. And due to the success of the pretrained language model BERT (Kenton and Toutanova, 2019), most of the recent work adopts the pretraining-finetuning paradigm, i.e., using a BERT-level pretrained model as backbone then fine-tune on specific datasets (Miftahutdinov and Tutubalina, 2019; Xu et al., 2020; Liang et al., 2021). This leads to the fact that we need to completely retrain the model when the standard termbase changes, which is not generalizable. Another bottleneck is that both mentions and terms in this task are short texts. Short text often contains insufficient information and introduces ambiguities, especially in the biomedical context, posing a huge challenge.

However new trends and solutions seem to have been presented to us in the era of Large Language Models (LLMs). The advanced embedding model has been regarded as a foundation model to be used for computing semantic similarity and retrieval, and the advanced models, such as instructor-xl (Su

et al., 2022), BGE (Xiao et al., 2023), and OpenAI's Text Embeddings (OpenAI, 2022, 2024). They are trained using effective methods as well as a large amount of supervised data and exhibit superior performance. Meanwhile, very Large language models seem to perform some kind of learning through the huge amount of data it has seen. Without gradient steps or fine-tuning, tasks can be accomplished simply from task definition and few-shot demonstrations provided within their contexts (Brown et al., 2020). This approach known as Language Prompting, or "Prompt" for short, has now become a new paradigm for accomplishing downstream tasks.

We intend to leverage the LLM and explore new paradigm-based solutions for the terminology normalization task. It is at this point when we were probing the LLM for the task of term normalization, we found that the LLM tends to understand and interpret the name of the mention or term that I entered, and we regarded it as a kind of informative expansion and could ease the problem of short texts. We provided this example in the Figure 1. Inspired by this we elaborate a format for knowledge acquisition, named knowledge card, which utilizes the knowledge and expands on the name of mentions or terms by knowledge distillation from LLM. Besides using the advanced embedding model, we use knowledge cards expanded from the LLM and propose a Knowledge-Enhanced Retrieval approach, which will consider both the name and the knowledge card in the retrieval. Experiments prove it is an effective unsupervised approach for the terminology normalization task.

Meanwhile, we found that ranking can also be realized by reasoning using the LLM, such as RankGPT Sun et al. (2023) has used the LLM to complete the work of ranking documents according to the user query. To further improve the performance, corresponding to the "recall and re-rank" framework, we propose a training-free framework for the terminology normalization task that leverages the capabilities of the advanced embedding models and LLMs.

Specifically, we use Knowledge-Enhanced Retrieval mentioned above as the rough recall module and design the "Top-k Ranking" module to accomplish this task using the LLM to further narrow down the range of candidate terms. Additionally, from the perspective of the professionalism of the normalization task, medical experts will follow protocols in annotation, which are not visible to us and vary from project to project, so we designed the "Protocol-Adaptive Re-ranking" module, try to use the LLM to discover the difference between protocols from the training data and use this as a basis for re-ranking candidate terms, to improve the accuracy and professionalism of the normalization conclusions. As shown in Figure 2, we show the overall framework and our contribution could be summarized as follows:

- We design a training-free framework for terminology normalization based on advanced embedding models and LLMs, to obtain the candidate terms via Knowledge-Enhanced Retrieval, and obtain the final standard terms through ranking with demonstration and chain-of-thought using a LLM.

- We propose a knowledge expansion approach that utilizes knowledge distilled from LLMs to extend short medical mentions and terms into knowledge cards containing enhanced descriptive information and medical knowledge.

- We have utilized prompt engineering techniques such as chain-of-thought instructions, demonstration selection, etc., to propose a workflow for ranking using LLM. Based on the idea of the Divide-and-Conquer algorithm, the "Top-K Ranking" module is used to further narrow down the candidate terms.

- We propose a "Protocol-Adaptive Re-ranking" module that uses the LLM to analyze the annotation protocols followed by experts from the training data to re-rank the candidate terms and also uses techniques such as ensemble to improve the accuracy and expertise of the normalization conclusions.

2 Related Work

2.1 Biomedical Terminology Normalization

Biomedical term normalization is one of the fundamental tasks within biomedical natural language processing (Leaman et al., 2013; Ji et al., 2020; Li et al., 2017), aiming at finding standard terms for a variety of different clinical statements.

Early approaches for clinical term normalization involve using dictionaries for lookup (Lee et al., 2016) or employing heuristic search methods based on string matching (Leal et al., 2015),

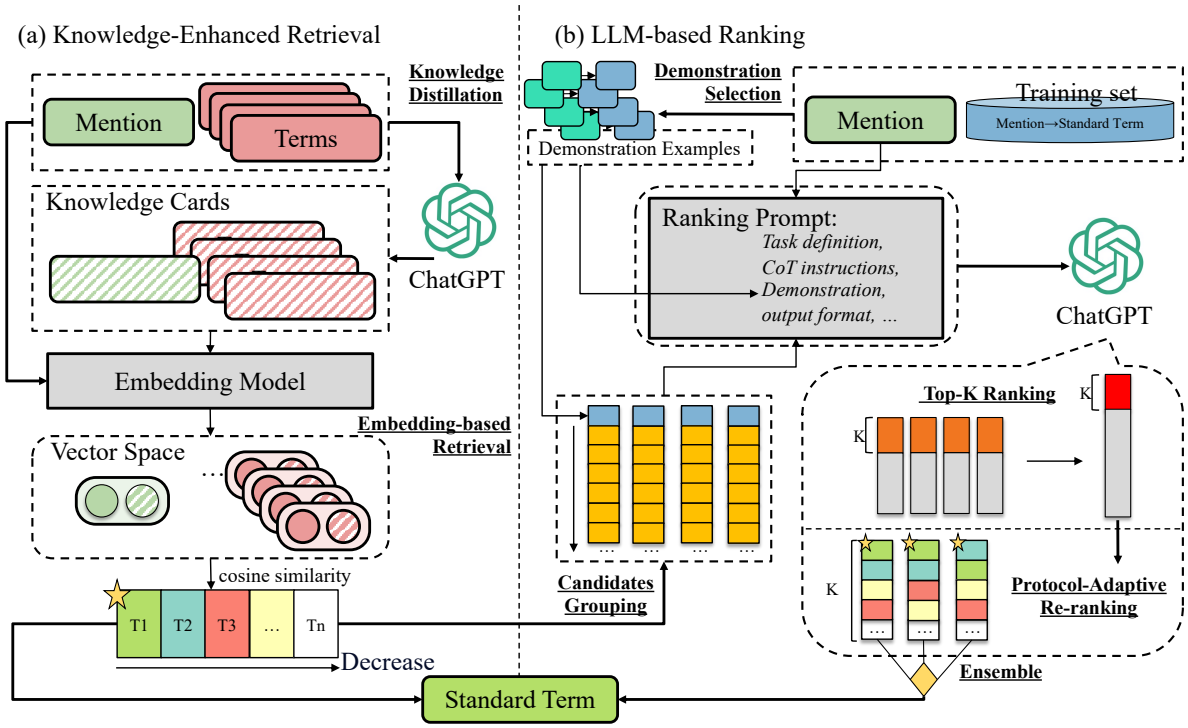

Figure 2: The proposed framework. The left side is the Knowledge-Enhanced Retrieval module, and the right side shows the LLM-based Ranking flow. The figure also shows two approaches that are cascaded for the terminology normalization task, where (a) is an unsupervised approach and (b) is an LLM-based supervised approach that follows (a).

which incurred significant manual effort. With artificial intelligence's advancement, machine learning and deep learning methods are increasingly emerging (Savova et al., 2008; Sui et al., 2022; Zhou et al., 2021b; Ji et al., 2021; Zhou et al., 2021a).

Due to the massive scale of the knowledge base, it becomes challenging to rank the entire standard terminology base directly. It is vital to recall some semantically related candidate terms for further ranking. Hence the two-stage clinical term normalization tasks involve two main steps: recall and rank, e.g., Liang et al. (2021) proposed a framework based on "recall, rank, and fusion," and introduced a model-based online negative sampling strategy in the recall stage. Xu et al. (2020) proposed an architecture consisting of a candidate generator and a list-wise ranker based on BERT.

The recall module could be traditional models such as elastic search, BM25, and TF-IDF, while vector-based text semantic similarity has become mainstream. Ji et al. (2020) first conducted the BM25 scores as the recall evaluation. Liu et al. (2020) provided an ABTSBM method for ICD-9-CM3 terminology normalization. The N-gram algorithm was used to generate a standard candidate terminology set. Niu et al. (2019) presented a multi-task character-level attentional network that learned character structure features. Yan et al. (2020) suggested a generative sequence framework to generate all the corresponding candidate medical procedure entities directly and adopt prefix tree decoding to avoid producing unrealistic results.

The ranking module is usually a scoring or classification model that incorporates various features to find the standard term corresponding to the mention from a few of candidates. Leaman et al. (2013) proposed a linear pair-wise model for the representation of medical terms, ranking standard terminologies based on the similarity between vectors and devising the strategies for choosing negative samples in the training process. In addition, many studies regard normalization tasks as a classification problem. Liu et al. (2020) use the BERT-based classification model to classify the correct standard terminology. Ji et al. (2020) fine-tuned the existing BERT models as well.

2.2 Leveraging Large Language Models

Recently, pretrained language models (Radford et al., 2018; Kenton and Toutanova, 2019) show promising improvements over many NLP tasks. Motivated by the finding that model scaling en-

Dataset	NAME	KC	HR@1	HR@5	HR@10	HR@20	HR@50	HR@100	HR@200
AskPatient	✓	✗	69.31	91.26	95.73	98.03	99.21	99.55	99.64
	✓	✓	**74.07**	**93.56**	**96.94**	**98.59**	**99.41**	**99.57**	**99.65**
TwADR-L	✓	✗	38.68	63.35	72.67	78.84	86.76	**91.38**	94.11
	✓	✓	**42.47**	**66.15**	**72.53**	**80.38**	**87.11**	91.10	**94.32**
SMM4H-17	✓	✗	55.92	74.60	82.56	88.92	93.32	95.36	96.48
	✓	✓	**64.40**	**80.28**	**87.00**	**91.12**	**94.96**	**96.00**	**96.76**

Table 1: The Knowledge-Enhanced Retrieval experiment result, where "NAME" denotes the names of mentions and terms used in retrieval, "KC" denotes the knowledge cards used in retrieval, "HR@num" denotes the hit rate of candidate terms containing the correct answer, and "num" denotes the number of candidate terms recalled.

hances the model capacity (Kaplan et al., 2020), the researchers explore the scaling effect further by scaling up the parameters to a larger size (Ouyang et al., 2022). With parameter scaling, LLMs exhibit some special and powerful abilities that allow for multiple ways to leverage LLMs to accomplish downstream tasks.

The concept of In-Context learning (ICL) is rigorously introduced by GPT-3 (Brown et al., 2020). This framework assumes that once the language model has been furnished with natural language instructions and multiple task demonstrations, it can generate the expected output of a test instance by completing the word order of the input text (prompt) without additional training or gradient updates (Zhao et al., 2023). For instance, through designing appropriate prompts, leveraging LLMs for knowledge acquisition becomes possible. Trajanoska et al. (2023) indicated that using advanced LLMs can improve the accuracy of the process of creating a Knowledge Graph from unstructured text. Nori et al. (2023) examines the impact of a range of prompting techniques on the performance of LLM in medicine, including chain-of-thought, kNN demonstration examples, and model output ensemble, which unleash top-performing specialist capabilities of LLM. RankGPT Sun et al. (2023) is exploring the use of large models to solve the problem of ranking related documents and exploring new paradigms for the task.

3 Method

We outline the comprehensive of our solution, it is a training-free framework based on LLM and comprises three primary modules: the "Knowledge-Enhanced Retrieval" module is to recall high-quality candidate terms and is also an advanced unsupervised normalization approach, the "Top-K Ranking" module and "Protocol-adaptive Re-ranking" module are to minimize the range of candidate terms and to find the optimal standard term, respectively, by using the LLM for ranking. Specific framework details are displayed in Figure 2.

3.1 Knowledge-Enhanced Retrieval

In this module there are two steps one is knowledge distillation, which uses a prompt to obtain the expanded information of mentions and terms from the LLM, and the second one is the embedding-based retrieval, which also utilizes the names of mentions and terms as well as the knowledge cards, obtains their knowledge enhanced vector representations and computes the semantic similarity to retrieval standard terms.

3.1.1 Knowledge Distillation

This step focuses on distilling knowledge from advanced LLM in the form of data using the language prompting technique, and then the knowledge is explicitly used to enhance the semantics of mentions and terms.

To begin with, we construct a seed task and craft a prompt manually. By utilizing the prompt engineering technique, we control the output format of LLM so that we can apply specific rules to clean the output. Specifically, we define the clear mission objectives and output formats and provide some reference dimensions. For instance, for a medicine term, the knowledge card contains pertinent details such as its definition description, active ingredient, content specification, dosage form, etc.

Also, the prompt contains some chain-of-thought instructions, which require the LLM to analyze the type of input mentions or terms, then refer to some dimensions given to determine the dimensions of this knowledge card, and finally output the specific content of the knowledge card according to the format and content requirements.

3.1.2 Embedding-based Retrieval

We employ "Embedding + Knowledge Card" as our final retrieval strategy, i.e., encode both the term name and its expanded information via knowledge cards as vectors by a text embedding model, concatenate them as the knowledge-enhanced representation for the term, and then compute the similarity score. The algorithm flow for this approach is presented in Algorithm 1. The vector retrieval engine embeds every standard term t in the standard terminology base T and its corresponding knowledge card K_t, and concatenates the term name embedding and knowledge card embedding into a vector $\hat{\mathbf{t}} \in \hat{\mathbf{T}}$. Meanwhile, the mention m, and its associated knowledge card K_m is encoded as $\hat{\mathbf{m}}$ through the same operation. The cosine similarities between the mention m and every standard term t in the entire terminology base are used as measures, some standard terms with high similarities to the mention m are selected and added to a candidate set C, and we select the term with the highest score as the standard term.

Algorithm 1: Algorithm of Knowledge-Enhanced Retrieval

Input: mention m
standard terminology base T
knowledge cards $K_m, K_t \in K_T$
Output: standard term s of mention m
candidate terms C of mention m

foreach t *in* T **do**
  embedToVecWithKC$(t, K_t) \rightarrow \hat{\mathbf{t}} \in \hat{\mathbf{T}}$;
end
embedToVecWithKC$(m, K_m) \rightarrow \hat{\mathbf{m}}$;
searchSimTerm$(m, T, \hat{\mathbf{m}}, \hat{\mathbf{T}}) \rightarrow C$;
searchMaxSimTerm$(m, T, \hat{\mathbf{m}}, \hat{\mathbf{T}}) \rightarrow s$;

3.2 LLM-based Ranking

To further improve performance, we proposed a training-free framework, which uses the previous unsupervised approach 3.1 as the rough recall model and then uses LLM to rank the recalled candidate terms even more finely.

3.2.1 Ranking Prompt Designing

One of the most important parts of using the LLM for downstream tasks is the design of the prompt including the system prompt for initial role and goal definitions, concentrating the capabilities of LLM on biomedical, and the content prompt for specific instructions, which focuses on the following five parts. Specific prompt content we provide in the appendix A.

The task definition for the LLM is to rank a given candidate terms list and then output the top K most relevant terms with the input mentions.

Chain-of-thought instructions are introduced for the LLM to perform step-by-step reasoning to improve the task accuracy, including learning the pattern from the given demonstrations, analyzing the meaning of the input mentions, giving the basis for this ranking and then outputting the ranking result.

Demonstrations have proven to be very effective information for LLM to conduct in-context learning to accomplish tasks. so we designed a demonstration selection module to find higher-quality demonstration examples from the training data based on the k-nearest neighbors algorithm. By calculating the similarity between the input mention and the mentions in training data, where the similarity is still based on the "Knowledge-Enhanced Retrieval" proposed above, based on the input mention m we find out the appropriate demonstration examples E from the training set D. The specific algorithm flow is shown in Algorithm 2.

Algorithm 2: Algorithm of Demonstration Selection

Input: given mention m
training dataset $(d, t) \in D$
knowledge cards $K_m, K_d \in K_D$
Output: k-NN demonstration examples E of input mention m

foreach d,_ *in* D **do**
  embedToVecWithKC$(d, K_d) \rightarrow \hat{\mathbf{d}} \in \hat{\mathbf{D}}$
end
embedToVecWithKC$(m, K_m) \rightarrow \hat{\mathbf{m}}$;
searchSimTrain$(m, D, \hat{\mathbf{m}}, \hat{\mathbf{D}}) \rightarrow E$;

Output format is an unnecessary part to realize a more automated and controllable algorithm process, we let the LLM output in JSON format so that it is easy to extract the conclusions and contents we want to obtain.

The task input is a mention and some candidate terms. Heuristically, we group the candidates so that the number of elements in each group stay at a suitable level. Moreover, discarding sequential grouping, we use a balanced grouping strategy that

randomly assigns candidates C to groups G according to their cosine scores, ensuring consistency in the number and distribution of each group. Since we have provided k-NN demonstration examples in our prompt, so we add the standard terms from these demonstration examples as expanded candidates to each group and obtain supplemented $\tilde{G}$.

3.2.2 Ranking and Re-ranking

The specific ranking procedure is that we finish a "Top-K Ranking" task, where the goal is to further filter the candidate terms, reducing the number to K, where K is a relatively small value. Then the "Protocol-Adaptive Re-ranking" module, re-ranks these terms and selects the most suitable standard term corresponding to the mention by the results ensemble after multiple re-rankings. The specific algorithm flow is shown in Algorithm 3.

Algorithm 3: Algorithm of LLM-based Ranking

Input: given mention m,
candidate terms set C
Output: nomalized result s

candidateGrouping$(C) \rightarrow g \in G$;
addDemocandidate$(G) \rightarrow \tilde{g} \in \tilde{G}$;
foreach $\tilde{g}$ *in* $\tilde{G}$ **do**
  topkRanking$(m, \tilde{g}) \rightarrow v \in V$;
end
topkRanking$(m, V) \rightarrow \tilde{C}$;
foreach n *in* $1, 2, ..., T$ **do**
  re-ranking$(m, \tilde{C}) \rightarrow r_n \in R$;
end
ensemble$(R) \rightarrow s$;

Top-K Ranking. Applying the divide-and-conquer algorithm, we find the top K terms from each group individually and combine the answers, and then find the top K terms v again from the new combination candidate set V, the final result is a set $\tilde{C}$ with only a small number of candidate terms.

Protocol-Adaptive Re-ranking. To find the most appropriate term from a smaller set of candidate terms $\tilde{C}$ as the standard term corresponding to the mention, we delete the constraint of finding K terms in the ranking prompt and change it to filtering out the relevant terms and then re-ranking them. Generally, different normalization task projects should have different annotation protocols when annotated by experts, thus we let the LLM consider discovering this kind of implicit information from the demonstrations and use it as the basis for re-ranking, to improve the normalization accuracy. Meanwhile, to make the best effort to eliminate the randomness of the final result, we use an ensemble strategy by re-ranking T times and then voting to get the final standard term s.

4 Experiment

4.1 Datasets

Following the complete setting of (Xu et al., 2020), We conduct our experiment on three datasets, AskPatient (Limsopatham and Collier, 2016), TwADR-L (Limsopatham and Collier, 2016), and SMM4H-17 (Sarker et al., 2018).

AskAPatient: The AskAPatient dataset[1] comprises 17,324 annotations of adverse drug reactions (ADRs) sourced from blog entries. These annotations are linked to 1,036 medical concepts, encompassing 22 semantic categories derived from a segment of the Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) and the Australian Medicines Terminology (AMT). Our methodology aligns with the 10-fold cross-validation framework utilized in the study by (Limsopatham and Collier, 2016), which presents 10 separate training, validation, and testing divisions.

TwADR-L: Encompassing 5,074 expressions of ADRs extracted from social media platforms, the TwADR-L dataset[1] aligns these expressions with 2,220 concepts from the Medical Dictionary for Regulatory Activities (MedDRA), spanning 18 semantic categories. Our approach here also adheres to the 10-fold cross-validation model established by (Limsopatham and Collier, 2016).

SMM4H-17: This dataset, SMM4H-17[2], includes 9,149 handpicked ADR expressions from Twitter posts. These expressions are linked to 22,500 concepts, incorporating 61 semantic types from MedDRA Preferred Terms (PTs). The training dataset includes 5,319 expressions from the publicly released set while reserving the 2,500 expressions from the original test set for evaluation purposes.

4.2 Implementation Details

For the Knowledge-Enhanced Retrieval, we use text-embedding-3-large (OpenAI, 2024) as our Em-

[1] https://zenodo.org/records/55013
[2] https://data.mendeley.com/datasets/rxwfb3tysd/1

Method	AskPatient	TwADR-L	SMM4H-17
Unsupervised methods			
TF-IDF	55.47	22.93	22.16
BM25	55.46	23.00	24.20
text-embedding-ada-002 (OpenAI, 2022)	64.94	35.18	45.48
text-embedding-3-large (OpenAI, 2024)	69.31	38.68	55.92
∗ text-embedding-ada-002 + KnowledgeCard	72.95	39.38	64.28
∗ text-embedding-3-large + KnowledgeCard	**74.07**	**42.47**	**64.40**
Supervised methods			
WordCNN (Limsopatham and Collier, 2016)	81.41	44.78	-
WordGRU+Attend+TF-IDF (Tutubalina et al., 2018)	85.71	-	-
BERT+TF-IDF (Miftahutdinov and Tutubalina, 2019)	-	-	89.64
CharCNN + Attend+MT (Niu et al., 2019)	84.65	46.46	-
CharLSTM + WordLSTM (Han et al., 2017)	-	-	87.20
LR + MeanEmbedding (Belousov et al., 2017)	-	-	87.70
BERT + BERT-rank + ST-reg (Xu et al., 2020)	87.46	47.02	88.24
∗ Ours	**88.54**	**52.28**	**90.84**

Table 2: Comparison of different approaches for biomedical terminology normalization. The evaluation metric is accuracy and the "∗" denotes our proposed approach or module.

bedding model, and we set the number of candidates as 200.

For the LLM-based Ranking part, we chose gpt-3.5-turbo-1106 (OpenAI, 2023) as the basic LLM, in the demonstration selection module, we chose 10 nearest-neighbor examples for each mention, in the candidates grouping step, we divided the 200 candidates into 4 groups by default, and in the "Top-K Ranking" module, we finally chose the top 10 terms as input candidates for the re-ranking module, and in the re-ranking module, the ensemble times is set to 3 by default. The temperature for LLM inference is set to 0 and the seed is set to 42.

4.3 Evaluation of Knowledge-Enhanced Retrieval

We conducted experiments to prove the importance of the knowledge card for the embedding-based retrieval stage, the evaluation metric is the Hit Rate, denoted as "HR@num", which means the ratio of samples in which the candidates contain the corresponding normalized term, where "num" represents the number of candidates to be retrieved, the results are displayed in the Table 1. Meanwhile, we found that this module can be used as an unsupervised approach to terminology normalization, so we also compared it to several unsupervised models, with the metric being accuracy, this result is displayed in the top half of the Table 2. Additionally in the demonstration selection module, as mentioned above we used the same retrieval technique for the selection of the demonstration examples and we show the corresponding effect in the Appendix Table A1.

It can be observed that in the recall phase, the results of all three datasets specify that the use of both mentions and the name of the term as well as the knowledge card will result in a higher hit rate than the use of only the name in general. Introducing knowledge cards enhances the retrieval process by incorporating additional information and context. This additional knowledge helps refine the candidate set and improves the recall rate.

Again, when we consider it as an unsupervised term normalization method, we only consider the term with the highest scores, and we still notice that the results after using the knowledge cards are much better than the traditional BM25 model and TF-IDF model, as well as better than just using the advanced embedding model.

These improvements are indicative of the fact that the introduction of knowledge cards can enhance the retrieval process by integrating additional information and context and that this additional knowledge helps the embedded vectors to have more specific semantics, helping to find terms that have the same semantics.

However, we have also noticed the superior performance of advanced embedding models, and it can be noted that when we select a larger number of candidates (e.g., 200), the difference between whether or not to use the knowledge card is not so significant, suggesting that these advanced models are learning richer semantics from a large amount

Setting	SMM4H-17
Top-K Ranking	HR@10
Ours	97.36
w/o Knowledge-Enhanced Retrieval	96.20
w/ Knowledge-Enhanced Retrieval	
w/o CoT Instructions	93.64
w/o Demonstration Examples	76.96
w/o Grouping	96.56
w/ Grouping	
w/o Balanced Grouping	97.12
w/o Expanded Candidates	93.04
Term Selection	Acc
Ours	90.84
w/o Knowledge-Enhanced Retrieval	90.64
w/ Knowledge-Enhanced Retrieval	
w/o CoT Instructions	84.92
w/o Demonstration Examples	58.40
w/o Grouping	90.72
w/ Grouping	
w/o Balanced Grouping	90.52
w/o Expanded Candidates	87.88
w/o Protocol-Adaptive Re-ranking	89.84
w/o Ensemble	90.47

Table 3: Ablation experiments to validate the effectiveness of individual modules, the indentation indicates the subordination between the different settings.

of data. In addition, in our demonstration selection experiments, we found that on the TwADR-L and SMM4H-17 datasets, sometimes the results are better without using the knowledge card instead, as we will discuss in the Limitation Section 6.

4.4 Evaluation of LLM-based Ranking

Although we proposed a training-free terminology normalization framework, we still make use of the demonstration examples from the training set to allow LLM to accomplish the task through in-context learning, and thus we compare our approach to supervised methods using the same datasets.

The evaluation metric of the final normalization result is the accuracy score, which denotes the percentage of samples where the selected term is the correct normalized term, and the bottom half of Table 2 presents the accuracy scores of the introduced methods compared to our proposed model. Meanwhile, to study the contribution of each module to the final result, we conducted ablation experiments on the SMM4H-17 dataset, which has the largest standard terminology base and the largest number of semantic types, the specific results of which are displayed in Table 3.

It can be observed that the effect of our proposed method is significantly improved over the models that have been fine-tuned on the individual datasets, but only to provide demonstration examples for in-context learning without the need for parameter fine-tuning. From the ablation experiments, it can be observed that all of our proposed modules contribute positively to the final performance, with the main contributing parts being the high-quality demonstration, the designed CoT instructions, the expanded candidate terms supplemented by the demonstration examples, and the protocol-adaptive re-ranking module. As the context lengths supported by current advanced LLMs have become longer and their logical reasoning has become more and more powerful, the grouping and ensemble strategies have turned out to be minor tricks, but have also had the effect of enhancing the robustness of the system.

5 Conclusion

In this paper, we propose a training-free biomedical normalization framework that leverages the advanced Embedding Model and LLM, which incorporates two key components, Knowledge-Enhanced Retrieval and LLM-based Ranking.

For Knowledge-Enhanced Retrieval, to eliminate the ambiguity caused by the short text, we expanded for mentions and terms using the LLM. We use both the distilled knowledge card from advanced LLM and the term name to get a more informative vector representation. It improves the accuracy and hit rate on different datasets without the additional training of a supervised recall model. It is also an unsupervised terminology normalization approach with significant improvement.

For LLM-based Ranking, We leverage the reasoning capabilities of the LLM to further rank and re-rank the candidate terms to improve performance. By designing a very complete and effective prompt, including task definition, CoT instructions, demonstration, output formatting requirements, etc., we leverage LLM to further narrow down the candidate words by completing the Top-K ranking task. After that, the number of candidate terms is reduced to K, which is a relatively small number. Then by modifying the prompt, we try to discover the protocols implicit in different datasets using the LLM and use them as influencing factors, and perform multiple re-rankings to get the final answer through ensemble learning. These protocols are followed by labeling experts and this professional knowledge can help improve performance.

6 Limitations

First of all, as mentioned above, we found that the knowledge cards showed a negative effect in the demonstration selection experiments, and by analyzing this we can see that we are calculating the semantic similarity between mentions when making the example selection, which is different from that between mentions and terms, which tend to have a slight difference in characters but the differences between mentions are not significant, especially because of the high repetition rate between mentions in the SMM4H-17 dataset. However, this also reflects that the knowledge card we distilled from the model is only a vague description of the knowledge of mentions or terms and not precise structured knowledge, and subsequent research can use this as an entry point to explore the interaction with LLM to distill more fine-grained knowledge.

Secondly, in the process of ranking using the large model, we found that some of the model outputs could not pass the format check, which might indicate that the model could not find the correct answer from the current candidates, and we dealt with the issue by choosing a more relaxed temperature, e.g., 0.5, which might have led to the incorrect delivery. But actually, using dynamic candidates could be a better solution. This also inspires us to follow up with multiple rounds of interactions with LLM to further improve the accuracy of the task.

Finally, We propose a training-free framework that leverages advanced LLMs such as ChatGPT to accomplish the task. However, even though we have set the temperature to 0 and provided fixed seeds, we still cannot eliminate its randomness, so there are a few potential risks, such as the fact that the knowledge cards obtained from distillation are rough and may contain hallucinatory or harmful information. It is worth mentioning that the probability of these occurrences is small and that has little impact on the performance of our approach.

References

Maksim Belousov, William G Dixon, and Goran Nenadic. 2017. Using an ensemble of linear and deep learning models in the smm4h 2017 medical concept normalisation task. In *SMM4H@ AMIA*, pages 54–58.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sifei Han, Tung Tran, Anthony Rios, and Ramakanth Kavuluru. 2017. Team uknlp: Detecting adrs, classifying medication intake messages, and normalizing adr mentions on twitter. In *SMM4H@ AMIA*, pages 49–53.

Zongcheng Ji, Qiang Wei, and Hua Xu. 2020. Bert-based ranking for biomedical entity normalization. *AMIA Summits on Translational Science Proceedings*, 2020:269.

Zongcheng Ji, Tian Xia, Mei Han, and Jing Xiao. 2021. A neural transition-based joint model for disease named entity recognition and normalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2819–2827, Online. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

André Leal, Bruno Martins, and Francisco M Couto. 2015. Ulisboa: Recognition and normalization of medical concepts. In *proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 406–411.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Hsin-Chun Lee, Yi-Yu Hsu, and Hung-Yu Kao. 2016. Audis: an automatic crf-enhanced disease normalization in biomedical text. *Database*, 2016:baw091.

Kahyun Lee and Özlem Uzuner. 2020. Normalizing adverse events using recurrent neural networks with attention. *AMIA Summits on Translational Science Proceedings*, 2020:345.

Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang. 2017. Cnn-based ranking for biomedical entity normalization. *BMC bioinformatics*, 18:79–86.

Ming Liang, Kui Xue, Qi Ye, and Tong Ruan. 2021. A combined recall and rank framework with online negative sampling for chinese procedure terminology normalization. *Bioinformatics*, 37(20):3610–3617.

Nut Limsopatham and Nigel Collier. 2016. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1014–1023.

Yijia Liu, Bin Ji, Jie Yu, Yusong Tan, Jun Ma, and Qingbo Wu. 2020. An advanced icd-9 terminology standardization method based on bert and text similarity. In *The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pages 1868–1879. Springer.

Yen-Fu Luo, Weiyi Sun, and Anna Rumshisky. 2019. Mcn: a comprehensive corpus for medical concept normalization. *Journal of biomedical informatics*, 92:103132.

Zulfat Miftahutdinov and Elena Tutubalina. 2019. Deep neural models for medical concept normalization in user-generated texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 393–399.

Jinghao Niu, Yehui Yang, Siheng Zhang, Zhengya Sun, and Wensheng Zhang. 2019. Multi-task character-level attentional networks for medical concept normalization. *Neural Processing Letters*, 49:1239–1256.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.

OpenAI. 2022. New and improved embedding model. Technical report.

OpenAI. 2023. New models and developer products announced at devday. Technical report.

OpenAI. 2024. New embedding models and api updates. Technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Patrick Ruch, Julien Gobeill, Christian Lovis, and Antoine Geissbühler. 2008. Automatic medical encoding with snomed categories. In *BMC medical informatics and decision making*, volume 8, pages 1–8. BioMed Central.

Abeed Sarker, Maksim Belousov, Jasper Friedrichs, Kai Hakala, Svetlana Kiritchenko, Farrokh Mehryary, Sifei Han, Tung Tran, Anthony Rios, Ramakanth Kavuluru, et al. 2018. Data and systems for medication-related text classification and concept normalization from twitter: insights from the social media mining for health (smm4h)-2017 shared task. *Journal of the American Medical Informatics Association*, 25(10):1274–1283.

Guergana K Savova, Anni R Coden, Igor L Sominsky, Rie Johnson, Philip V Ogren, Piet C De Groen, and Christopher G Chute. 2008. Word sense disambiguation across two domains: biomedical literature and clinical notes. *Journal of biomedical informatics*, 41(6):1088–1100.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Xuhui Sui, Kehui Song, Baohang Zhou, Ying Zhang, and Xiaojie Yuan. 2022. A multi-task learning framework for chinese medical procedure entity normalization. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8337–8341. IEEE.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*.

Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing knowledge graph construction using large language models.

Elena Tutubalina, Zulfat Miftahutdinov, Sergey Nikolenko, and Valentin Malykh. 2018. Medical concept normalization in social media posts with recurrent neural networks. *Journal of biomedical informatics*, 84:93–102.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Dongfang Xu, Zeyu Zhang, and Steven Bethard. 2020. A generate-and-rank framework with semantic type regularization for biomedical concept normalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8452–8464.

Jinghui Yan, Yining Wang, Lu Xiang, Yu Zhou, and Chengqing Zong. 2020. A knowledge-driven generative model for multi-implication chinese medical procedure entity normalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1490–1499.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Baohang Zhou, Xiangrui Cai, Ying Zhang, Wenya Guo, and Xiaojie Yuan. 2021a. Mtaal: multi-task adversarial active learning for medical named entity recognition and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14586–14593.

Baohang Zhou, Xiangrui Cai, Ying Zhang, and Xiaojie Yuan. 2021b. An end-to-end progressive multi-task learning framework for medical named entity recognition and normalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6214–6224, Online. Association for Computational Linguistics.

A The Specific Prompts

Here are the specific contents of the prompts used in this article, including the prompt for knowledge card generation, "Top-K Ranking" and "Protocol-Adaptive Re-Ranking", they are shown in Figure A1, Figure A2 and Figure A3.

Dataset	De-dup	NAME	KC	HR@1	HR@5	HR@10	HR@20	HR@50	HR@100	HR@200
AskPatient	✗	✓	✗	82.68	93.98	96.21	97.65	98.95	99.51	99.71
		✓	✓	**84.30**	**94.26**	**96.36**	**97.67**	**99.00**	**99.56**	**99.81**
	✓	✓	✗	70.92	89.65	93.47	95.96	98.19	99.15	99.50
		✓	✓	**73.67**	**90.10**	**93.75**	**96.01**	**98.27**	**99.24**	**99.68**
TwADR-L	✗	✓	✗	**41.20**	73.70	81.87	87.83	**93.11**	95.79	**97.93**
		✓	✓	40.06	**74.47**	**82.72**	**88.30**	93.04	**96.10**	97.63
	✓	✓	✗	**25.85**	**60.18**	71.54	80.92	89.00	93.33	**96.69**
		✓	✓	23.40	59.60	**72.47**	**81.27**	**88.87**	**93.75**	96.19
SMM4H-17	✗	✓	✗	**89.68**	**94.96**	**96.20**	**97.16**	**97.72**	**97.88**	**98.08**
		✓	✓	89.48	94.72	96.08	96.80	97.36	97.84	98.00
	✓	✓	✗	**68.95**	**84.84**	**88.56**	**91.46**	**93.14**	**93.62**	**94.22**
		✓	✓	68.35	84.12	88.21	90.37	92.06	93.50	93.98

Table A1: The Demonstration Selection experiment, where "De-dup" denotes deduplication, meaning that I remove samples in the test set that duplicate mentions in the training set, "NAME" denotes the names of mentions and terms used in retrieval, "KC" denotes the knowledge cards used in retrieval, "HR@num" denotes the hit rate of the terms of examples containing the standard term corresponding to the input mention, and "num" denotes the number of examples recalled.

user:
You are asked to play the role of a doctor and you need to help me with a knowledge card generation task based on your medical knowledge.
For knowledge Card Generation, please recognize the medical terms in the input (e.g., disease, symptom, procedure, medication) and generate a knowledge card for them.
Please decide on the content of the knowledge card based on your medical knowledge, but it must include definitional descriptions and I will give you some references for common terminology type content. Knowledge card content needs to be exported item by item.

Knowledge Card Content Dimension Reference:
Disease diagnosis terms can contain dimensions such as definition description, etiology, pathology, site, disease type, and clinical manifestations (e.g., symptoms, characteristics, classification, gender, age, acute chronic, onset time).
Symptom terms may contain dimensions such as definition description, cause, classification, site, characteristics, and associated diseases.
Surgical operation terms may contain dimensions such as definition description, surgical technique, target site, surgical approach, and nature of the surgical condition, etc.
Medicine terms can contain dimensions such as definition description, active ingredient, content specification, dosage form, etc.

Requirements:
1. be as detailed as possible, consistent with medical knowledge, not made up, unrecognized term types and dimensions need not be output.
2. do not refuse to answer, output relevant medical knowledge as much as possible.
3. indicate the type of terminology, if possible
4. do not engage in explanations and politeness.
5. do not make additional summaries.

Input:
{term}

Knowledge Card:

Figure A1: The specific prompt for knowledge card generation, used in the knowledge distillation step of the Knowledge-Enhanced Retrieval.

system:
You are TermRankGPT, an intelligent assistant that ranks the input terms based on their semantic similarity to the meaning of the input mention. The more semantically similar, the higher the ranking. Note that mentions are often written in an informal way and terms are written in a relatively formal way.

user:
I will provide you with several candidate terms, your task is to output the most relevant topk terms after your ranking, in this task k is set to 10.

I have also provided some examples of mention with its corresponding standard term annotated by experts and some special cases.
[Example]:
{example}

[Two Special Cases]:
1. If the mention input is the same as a term, this term should be put at the top of the ranking topk_list.
2. If the mention in the examples are the same as the input mention, the corresponding term in the example should be put at the top of the ranking topk_list.

Follow the steps below for step-by-step reasoning:
1. Summarize the correspondence between mentions and terms from examples as the ranking reference.
2. Analyze the meaning of the input mention or the state it describes.
3. Give the basis for this ranking.
4. Rank the candidate list and select the topk terms according to the task objectives.
5. Final check: Determine if there are any special cases I mentioned before, if so, correct the ranking result.

Please follow the above reasoning steps for the task input and then output the reasoning process and and the selected topk terms in the follow JSON format::
{
"reasoning_process": 1.xxx, 2.xxx, ...,
"topk_list": [term1,term2,...] ,
}

[Task Input]:
mention:
{mention}

List of candidate terms:
{cand}

[Task Output]:

Figure A2: The specific prompt for “Top-K Ranking” task.

system:
You are TermRankGPT, an intelligent assistant that ranks the input terms based on their semantic similarity to the meaning of the input mention. The more semantically similar, the higher the ranking. Note that mentions are often written in an informal way and terms are written in a relatively formal way.

user:
will provide you with several candidates, your task is to find the term that is closest to its meaning or to the state it describes for the input mention as its standard term from the input candidates, and then re-rank candidate list according to the task objectives.

I have also provided some examples of mention with its corresponding standard term annotated by experts and some special cases.
[Example]:
{example}

[Three Special cases]:
1. If the mention input is exactly the same as one term, this term should be put at the top of the ranking result list.
2. If the mention in the examples is exactly the same as the input mention, the corresponding term in the example should be put at the top of the ranking result list.
3. If more than one standard terms are selected the annotation preferences and habits of the experts should be considered in ranking.

Follow the steps below to reason about the task input step by step, giving details of the process at each step::
1. Summarize the correspondence between mentions and terms and the annotation preferences and habits of experts from examples as the ranking reference.
2. Analyze the meaning of the input mention or the state it describes.
3. Give the basis for this ranking.
4. Rank the selected terms according to the task objectives.
5. Final check: Determine if there are any special cases I mentioned before, if so, correct the ranking result.

Please follow the above reasoning steps for the task input and then output the reasoning process and ranking result in format as follows, note that the ranking result is in JSON format::
{
"reasoning_process": 1.xxx, 2.xxx, ...,
"ranking_result": [term1, term2, …]
}

[Task Input]:
mention:
{mention}

List of candidate terms:
{cand}

[Task Output]:

Figure A3: The specific prompt for “Protocol-Adaptive Re-ranking” task.