MT-PATCHER: Selective and Extendable Knowledge Distillation from Large Language Models for Machine Translation

Jiahuan Li[♣*], **Shanbo Cheng**[♠†], **Shujian Huang**[♣†] and **Jiajun Chen**[♣]
[♣] National Key Laboratory for Novel Software Technology, Nanjing University, China
[♠] ByteDance Research
lijh@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn
chengshanbo@bytedance.com

Abstract

Large Language Models (LLM) have demonstrated their strong ability in the field of machine translation (MT), yet they suffer from high computational cost and latency. Therefore, transferring translation knowledge from giant LLMs to medium-sized machine translation models is a promising research direction. However, traditional knowledge distillation methods do not take the capability of student and teacher models into consideration, therefore repeatedly teaching student models on the knowledge they have learned, and failing to extend to novel contexts and knowledge. In this paper, we propose a framework called MT-PATCHER, which transfers knowledge from LLMs to existing MT models in a *selective, comprehensive and proactive* manner. Considering the current translation ability of student MT models, we only identify and correct their translation errors, instead of distilling the whole translation from the teacher. Leveraging the strong language abilities of LLMs, we instruct LLM teachers to synthesize diverse contexts and anticipate more potential errors for the student. Experiment results on translating both specific language phenomena and general MT benchmarks demonstrate that finetuning the student MT model on about 10% examples can achieve comparable results to the traditional knowledge distillation method, and synthesized potential errors and diverse contexts further improve translation performances on unseen contexts and words.

1 Introduction

Large Language Models (LLM) have shown their impressive capabilities across almost all natural language tasks (Brown et al., 2020; Zhao et al., 2023). However, their ability strongly correlates with the model size. In the field of machine translation, competitive results can only be evidenced on larger LLMs, while medium-sized LLMs like Alpaca (Taori et al., 2023) and ParroT (Jiao et al., 2023a) still lag behind supervised NMT systems by a large margin (Jiao et al., 2023a; Zhu et al., 2023). How to efficiently transfer knowledge from larger LLMs to existing MT models that are affordable to deploy, is an important research direction.

The most common method for knowledge transferring is knowledge distillation (KD) (Hinton et al., 2015; Kim and Rush, 2016), where given an unlabeled corpus, a student model is trained to mimic the output of a teacher model on the corpus. Although KD is a well-studied technique and has proven effective in many previous works (Kim and Rush, 2016; Wang et al., 2021; Liu et al., 2023), we argue that when transferring knowledge from giant LLMs to existing MT models, the traditional KD method does not take the capability of the student and teacher model into consideration, therefore leaving much room for improvement in terms of both efficiency and effectiveness.

Firstly, in contrast to student models in previous works (Kim and Rush, 2016; Wang et al., 2021; Liu et al., 2023) that are randomly initialized, recent student MT models (Hsieh et al., 2023; Fu et al., 2023) already exhibit a reasonable level of language proficiency, i.e., they can already accurately translate most examples in the unlabeled corpus. This renders the fine-tuning of student models on *all* teacher outputs both redundant and inefficient.

Secondly, the efficacy of KD is significantly constrained by the coverage of the monolingual corpus, which impedes their performance when translating words in novel contexts or words unseen in the monolingual corpus. However, modern LLMs grasp strong translation and language knowledge, as well as the ability to follow human instructions. This enables the development of more efficient and effective strategies for addressing these problems.

* Work done while Jiahuan Li's internship at ByteDance Research.
† Corresponding authors.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6445–6459
June 16-21, 2024

In this paper, we introduce MT-PATCHER, a novel framework designed for the knowledge transfer from LLMs to existing MT models in a *selective, comprehensive, and proactive* manner. The design philosophy of MT-PATCHER is inspired by effective teaching strategies observed in real-world scenarios. Rather than subjecting students to endless drills, an effective teacher would first assess the student's current abilities, then design practice to reinforce areas of weakness and extend learning to new situations (Lee Jr and Pruitt, 1979; Epstein and Voorhis, 2001). Leveraging the strong language capabilities of LLMs, our method seeks to emulate these pedagogical strategies. Specifically, we gather instructional data from GPT-4, which demonstrates how to identify and correct errors in student model translations, anticipate additional potential errors that the student models may commit, and synthesize diverse contexts for relevant translation knowledge that aids the student model in rectifying these errors. We subsequently finetune an existing proficient LLM on these data to transform it into an MT-PATCHER model.

We conduct experiments on translating specific language phenomena (chemistry materials and Chinese idioms) and on general machine translation benchmarks (WMT22 Chinese → English, English → German and English → Japanese). Experimental results show that finetuning the student model on only 10% examples selected by MT-PATCHER is equivalent to finetuning on all examples as in KD, and enlarging the finetuning corpus via the context synthesis and proactive error prediction technique further improves the translation performance.

2 Background

Large Language Model for Machine Translation Numerous studies have attempted to leverage LLMs for machine translation. Initial efforts (Lin et al., 2022; Vilar et al., 2022; Agrawal et al., 2023; Zhu et al., 2023; Hendy et al., 2023; Jiao et al., 2023b) centered on in-context learning, which utilizes several translation examples to guide the translation behavior of LLMs. Subsequent research (Jiao et al., 2023a; Li et al., 2023) shifted the focus to fine-tuning LLMs on existing parallel corpora to more effectively harness their translation capabilities. However, the translation performance of LLMs has not been as remarkable as their performance in other NLP tasks. Only state-of-the-art LLMs such as GPT-3 and GPT-4, which boast more than 100 billion parameters, can rival the performance of commercial translation systems (Hendy et al., 2023; Jiao et al., 2023b). Meanwhile, other medium-sized LLMs significantly trail behind supervised MT models (Zhu et al., 2023; Li et al., 2023; Jiao et al., 2023a). Li et al. (2023) suggest that the primary barrier to enhancing LLMs' performance is the lack of translation knowledge. Given that larger LLMs inherently possess more knowledge due to the scaling law (Kaplan et al., 2020), our work concentrates on transferring knowledge from these models to existing MT models.

Knowledge Distillation for Neural Machine Translation Knowledge distillation (KD), which improves smaller *student* models by learning on larger *teacher* models' output, is widely used in machine translation. Two common KD methods are LogitKD (Hinton et al., 2015; Tan et al., 2018), which optimizes the student model to match the teacher model's predicted distribution, and Sequence KD (SeqKD) (Kim and Rush, 2016; Wang et al., 2021; Gu et al., 2018; Zhou et al., 2019), where the student learns from the teacher-generated pseudo target sequence. As LogitKD requires access to the teacher's logits, it is impractical for distilling from proprietary LLMs. Therefore, we base our method on SeqKD, where student refers the smaller MT model we would like to improve, and teacher refers to larger LLMs which possess more translation knowledge than student.

Selective KD has been proposed by Wang et al. (2021) and Liu et al. (2023), but they all rely on comparing student models' outputs to oracle references. Unlike these works, our method instructs the LLM to identify student translation errors directly.

Large Language Model for Synthesizing Datasets With the growing generative capabilities of Large Language Models (LLMs), many works attempt to harness them for corpora generation. The generated corpora can serve as demonstrations for few-shot prompting (Sahu et al., 2022), fine-tuning corpora for existing models (Yoo et al., 2021), or seed corpora for human refinement (Yuan et al., 2021a). Studies such as Chung et al. (2023); Yu et al. (2023) also explore ways to balance diversity, accuracy, and bias reduction in LLM-based dataset synthesis. However, these approaches often generate datasets from scratch, ignoring the capabilities of the models being optimized, resulting in less efficiency compared to our method.

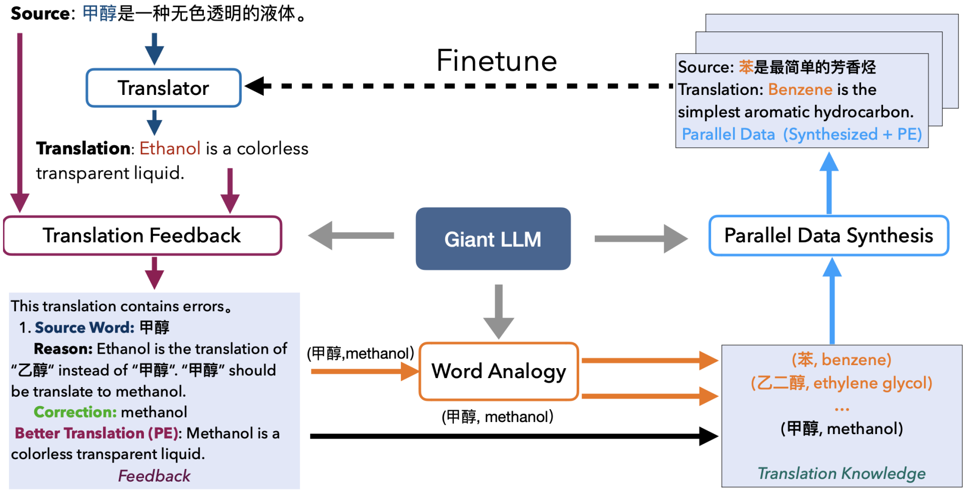

Figure 1: The illustration of MT-PATCHER framework. The correct translation for the source sentence should be 'Methanol is a colorless transparent liquid.'.

3 Methodology

In this section, we present MT-PATCHER , a framework that distills knowledge from LLMs to existing MT systems more efficiently and effectively. The process of MT-PATCHER undergoes two stages:

- **Knowledge Selection**: In this stage, the LLM acts as the *feedbacker*, which provides natural language feedback to translations of student models. Based on the feedback, we select source sentences with identified errors, which indicate knowledge deficiency of the student models, to the next stage.
- **Knowledge Extension**: In this stage, the LLM acts as the *parallel data synthesizer* and *word analoger*, which help the student model learn words it makes mistakes on by extending to more diverse contexts and similar words.

Figure 1 illustrates how MT-PATCHER works.

3.1 Knowledge Selection via *Feedbacker*

When transferring knowledge from LLMs to existing MT models, traditional SeqKD would finetune the student model on *all* teacher's output, ignoring the fact that the student model can already translate most of the examples well. Furthermore, several recent studies have unveiled emergent abilities in LLMs, such as *Self-Refinement* (Madaan et al., 2023) and *Self-Debug* (Chen et al., 2024), suggesting that iterative refinement of an initial draft may be a more effective strategy to tap into the knowledge reserves of LLMs.

To improve the efficiency of SeqKD and better elicit LLMs' knowledge, we propose to finetune LLMs to be a *feedbacker*, which produces natural language feedback of the student models' translation instead of directly generating its own translations. Formally, given a source sentence X and its corresponding translation Y, the goal of the feedbacker is to generate a comprehensive assessment f. This assessment comprises tuples of $(c, \{(s_i, e_i, t_i)\}_{i=1}^{N}, p)$, where c describes whether Y contains translation errors, s_i, e_i, t_i corresponds to the source span, explanation and correction of the i-th identified error, respectively, and p is the final post-edited translation that incorporates all error corrections.

3.2 Knowledge Extension via *Parallel Data Synthesizer and Word Analoger*

Another limitation of SeqKD is that the knowledge it can transfer is strictly limited to the given monolingual corpus. This limitation can hinder its generalizability in two key ways. Firstly, the correct translation of mistranslated words or phrases can only be learned within the contexts present in the given monolingual corpus, potentially limiting its applicability to broader contexts. Secondly, SeqKD also lacks the capacity for knowledge extrapolation, which prevents it from transferring knowledge that does not occur in the monolingual corpus.

Inspired by the principle of knowledge extension when designing good practice in the educational process (Lee Jr and Pruitt, 1979; Epstein and Voorhis, 2001), we transform LLMs into two modules to mitigate above two problems, respectively:

parallel data synthesizer and *word analoger*.

Parallel Data Synthesizer The goal of the parallel data synthesizer is to synthesize parallel sentences (X', Y') that contain a specific pair of phrases (s, c) where the student model makes mistakes in the context (X, Y), in order to generalize the current translation knowledge to more contexts. Ideally, the synthesized parallel sentences should be semantically diverse yet still similar to the original context in other aspects. However, in the preliminary experiments, we find that even for powerful LLMs like GPT-4, when conditioning them on the original context (X, Y), the generated parallel data lacks diversity and mostly resembles (X, Y).

To tackle this problem, we introduce another module called sentence analyzer, which first extracts the information of *domain, topic* and *style* of the original context. We then instruct the LLMs to synthesize parallel sentences with the same attributes as well as containing the phrase pair (s, c). This process can be seen as an information bottleneck where we squeeze the semantic information yet keep other attributes.

Word Analoger We further introduce the word analoger to proactively predict potential errors the student model may commit. For example, if the student MT model incorrectly translates the term *methanol*, an educated guess is that it may struggle with translating words within the domain of chemistry, such as *benzene* and *ethanol*. By anticipating these potential errors, we can enhance the student model's translation capability for words not present in the monolingual corpus.

Practically, given a source sentence X and a word s that the student MT model mistranslates, the word analoger aims to associate more words from two perspectives: (1) category, i.e., words belonging to the same category as s, and (2) semantic, i.e., words that frequently co-occur with s. We also require that the generated words should be rare and challenging in the prompt, ensuring that the student model will struggle to translate them accurately.

3.3 Implementation of MT-Patcher

Theoretically, state-of-the-art LLMs like GPT-4 can already serve as an MT-Patcher to transfer its knowledge to MT models. However, in practice, because we do not have unlimited access to GPT-4, we instead collect the demonstration data from GPT-4. Specifically, given a student model, we first use it to generate its translation on 20,000 monolingual sentences randomly selected from the monolingual corpus. We then leverage GPT-4 to execute the pipeline of MT-Patcher including (1) giving feedback f given the source sentence and student's translation (X, Y), (2) analyzing the domain, topic and style (d, t, st) of the source sentence X (3) making analogies (WA_x, WA_y) given the source sentence X and a word s in X (4) synthesizing parallel sentences containing error source words s and their corrections c with the same domain, topic and style attribute (d, t, st). Finally, we finetune the teacher LLM on these data to transform it to an MT-Patcher. All prompts we use for building MT-Patcher can be found in Appendix A.

4 Experiments

We evaluate our method on Chinese $\rightarrow$ English and English $\rightarrow$ German translation.

4.1 Experimental Settings

Student Translation Model For student translation models, we consider NLLB-200 3.3B (NLLB Team et al., 2022), a multilingual translation model pre-trained on 200 languages. Having been trained on massive parallel data, it can already translate reasonably well but falls short of language knowledge compared to LLMs, making it an ideal knowledge recipient for our experiment.

Due to the increasing interest in adopting LLMs for MT, we also consider ParroT (Jiao et al., 2023a), an LLM-based MT model finetuned on WMT validation sets from LLaMA-7B (Touvron et al., 2023).

Backbone LLM for MT-Patcher The backbone LLMs for building MT-Patcher in this paper are LLaMA2-13B (Touvron et al., 2023) and Baichuan-2-13B (Baichuan Inc, 2023). LLaMA2-13B is an English LLM and used to build MT-Patcher for English-German translation models. Baichuan-2-13B is trained on a mix of both Chinese and English corpus and demonstrates much stronger abilities in Chinese compared to LLaMA2. Therefore, we adopt it for building MT-Patcher for Chinese-English translation models. For each language pair considered, we fully finetune the corresponding LLM on the collected data for 3 epochs. See Appendix B for more implementation details.

Competitors We compare the translation performance of the following methods:

System	Chinese → English *Teacher Model: Baichuan2 13B*				English → German *Teacher Model: Llama2 13B*			
	$\lvert\mathcal{D}_f\rvert$	COMET	BLEURT	BLEU	$\lvert\mathcal{D}_f\rvert$	COMET	BLEURT	BLEU
Teacher	-	80.5	67.8	23.9	-	81.4	72.9	26.0
Student Model: ParroT-7B								
Student	-	75.4	60.6	18.1	-	80.5	69.0	23.9
SeqKD-Equal	119k	76.0	61.4	21.9	107k	80.3	70.8	24.1
SeqKD-Full	1M	76.5	61.7	22.2	1M	80.9	71.4	24.6
MT-PATCHER								
+ PE	119k	76.7	61.8	22.4	107k	80.9	71.6	24.9
+ PE + PDS	595k	77.4	62.6	23.0	535k	81.3	72.0	25.5
+ PE + PDS + WA	1.07M	**78.2**	**63.5**	**23.8**	963k	**81.8**	**72.6**	**26.2**
Student Model: NLLB 3.3B								
Student	-	76.8	63.9	20.8	-	86.1	76.3	34.3
SeqKD-Equal	104k	79.1	66.3	25.0	124k	85.2	74.7	32.0
SeqKD-Full	1M	79.5	66.9	**25.5**	1M	84.8	74.1	31.2
MT-PATCHER								
+ PE	104k	79.4	67.0	24.2	87k	86.2	76.5	34.5
+ PE + PDS	520k	79.9	67.4	24.8	435k	86.5	77.0	34.9
+ PE + PDS + WA	936k	**80.3**	**68.1**	25.4	783k	**87.2**	**77.5**	**35.6**

Table 1: Translation performance of the proposed method and other baselines on the WMT22 Chinese→English and English→German test sets. $|D_f|$ denotes the number of examples used to finetune the student model. SeqKD-Full refers to the student model finetunes on the full 1M pseudo parallel sentences, while SeqKD-Equal finetunes on random subsets of the teacher's translations with equal size to that of MT-PATCHER.

- **Student** is the translation model to be patched. In this paper, it refers to NLLB 3.3B or ParroT.

- **Teacher** is the model that is achieved by finetuning the larger LLM to perform translation directly. For a fair comparison, we finetune the LLM on GPT-4's translation on the monolingual sentences.

- **SeqKD** are models achieved by finetuning the Student model on the Teacher's translations.

- **MT-PATCHER (PE)** is the variant of MT-PATCHER , finetuning the Student model on the post-editing results in feedback.

- **MT-PATCHER (PE + PDS)** is the variant of MT-PATCHER which finetunes the Student model on the post-editing results as well as additional synthesized parallel sentences generated by parallel data synthesizer containing (error, correction) pairs. Unless other stated, we set the number of pseudo-parallel sentences to be 4 in this paper.

- **MT-PATCHER (PE + PDS + WA)** is the variant of MT-PATCHER which finetunes the Student model on the post-editing results and parallel sentences generated by parallel data synthesizer containing (error, correction) pairs and additional word pairs from word analoger. We generate 2 analogous words for each category and 1 context for each word.

4.2 Results on General Machine Translation

Table 1 presents experimental results on general machine translation benchmarks: WMT22 Chinese→English and English→German translation. We randomly select 1,000,000 sentences from RefinedWeb (Penedo et al., 2023) and WuDao 2.0 (Yuan et al., 2021b), respectively, as English and Chinese monolingual corpus. Performance are evaluated in COMET (Rei et al., 2020), BLEURT (Sellam et al., 2020) [1] and sacreBLEU (Post, 2018). We can see that:

MT-PATCHER can select more valuable examples. From Table 1, we can first see that the performance of MT-PATCHER (PE) is better SeqKD-Equal, and can be comparable to SeqKD-Full. This indicates the proposed method can select more valuable examples and discard useless examples. We also find our method suffers less from catastrophic forgetting compared to SeqKD-Full (See Appendix C for more experimental results). This makes MT-PATCHER an appealing method for real-world applications, considering the cost for finetuning the Student model is growing nowadays.

[1]The model we used for COMET and BLEURT is wmt22-comet-da and BLEURT-20, respectively.

	Chemistry Materials				Chinese Idioms			
	Unseen Context		*Unseen Word*		*Unseen Context*		*Unseen Word*	
	Accuracy	Rel. Perf.	Accuracy	Rel. Perf.	Score	Rel. Perf.	Score	Rel. Perf.
Student	6.0	22.4%	6.3	23.7%	1.20	39.8%	1.16	37.4%
Teacher	26.0	97.4%	25.8	97.4%	2.78	92.3%	2.82	91.0%
Feedbacker	26.7	100%	26.5	100%	3.01	100%	3.10	100%
SeqKD-Full	15.5	58.1%	10.6	40.0%	1.65	54.8%	1.62	52.3%
MT-PATCHER								
+ PE	15.8	59.2%	11.0	41.5%	1.73	57.5%	1.78	57.4%
+ PE + PDS	21.4	80.5%	11.2	42.3%	2.04	67.8%	1.81	58.4%
+ PE + PDS + WA	21.9	82.0%	16.3	61.5%	2.10	69.8%	2.02	65.2%

Table 2: Performance of different models when translating chemistry materials (evaluated in accuracy) and Chinese Idioms (evaluated by scores given by GPT-4). Rel. Perf: the relative performances of models compared to feedbacker, which is the best extent we can elicit knowledge from LLMs in this table.

	BLEU	COMET	BLEURT
Student	15.4	85.1	58.6
SeqKD	16.3	85.7	61.6
MT-PATCHER	**16.8**	**86.4**	**62.2**

Table 3: Effectiveness of MT-PATCHER on WMT English $\rightarrow$ Japanese translation test sets. The student model is NLLB 3.3B.

Parallel data synthesizer and word analoger improve the effectiveness of MT-PATCHER. We can also see that applying the parallel data synthesizer and word analoger to generate more patch data can further improve the translation performance of MT-PATCHER, highlighting the benefits of extending coverage of context and knowledge during the process of knowledge transferring.

It is worth noting that in the English $\rightarrow$ German direction, the teacher based on LLaMA-2-13B performs substantially worse than the student (NLLB 3.3B), which is consistent with previous findings (Li et al., 2023) that it is not trivial to adopt existing LLMs to outperform supervised translation models. As a result, SeqKD from this teacher leads to poor performance. However, based on the same backbone LLM, MT-PATCHER can still improve the performance of the Student model. This can be attributed to the hypothesis that revising an initial draft is a better way to elicit the knowledge of LLMs than direct generation, which we provide a further analysis in Section 5.2.

MT-PATCHER also works when the teacher is not very strong. Although we mainly focus on settings where we have strong teachers (which is why we choose different teacher models for English $\rightarrow$ German and Chinese $\rightarrow$ English translation), we also experiment with medium resource translation: WMT22 English $\rightarrow$ Japanese, using LLaMA2 as the teacher and NLLB 3.3B as the student. We present the results in Table 3. We find MT-PATCHER can still outperform SeqKD in this setting.

4.3 Results on Specific Language Phenomena

In order to understand how MT-PATCHER can improve the effectiveness of knowledge transfer, we present experiments on the Chinese-to-English translation for two specific language phenomena: *chemistry materials* and *Chinese idioms*. We select them for two reasons: **(1)** Both belong to long-tailed knowledge that student MT models cannot grasp very well. **(2)** There are also distinctions between them: chemistry materials represent simple, context-free knowledge, while Chinese idioms represent more abstract and metaphorical knowledge.

Specifically, for each language phenomenon, we first collect a list of 6,000 of them and their corresponding translations from the web. We then split these word pairs into two categories: *Seen* and *Unseen*, and create a monolingual set as well as two test sets based on the split [2]:

- **Monolingual Set**. For each word pair in the *Seen* set, we ask GPT-4 to synthesize one sentence that contains the source word. This set is for SeqKD and MT-PATCHER to leverage.
- **Test Set for Unseen Context.** For each word pair in the *Seen* set, we also ask GPT-4 to synthesize one parallel sentence pair that contains the source and target word in the source and target sentence, respectively. This set is for testing models' generalization ability when source words are seen yet contexts are novel.

[2] Details of the dataset and data split can be found in Appendix D.

- **Test Set for Unseen Word.** We collect the test set for Unseen Word in a similar way as Unseen Context using the word pairs in the *Unseen* set. This set is for testing models' generalization ability to novel words.

We take the Baichuan-2-13B as the LLM and NLLB 3.3B as the student model, and present the experimental results in Table 2. The accuracy of translating chemistry materials represents the percentage of test examples where the correct translation of the source chemistry material is found in the translation. Regarding Chinese idioms, due to the difficulty of providing reference translations of them, we instead ask GPT-4 to assess the translation quality given the source sentence, target sentence and dictionary definition. We report the average score, which ranges from 0 to 5. For ease of comparison, we also report how different models perform relative to the feedbackers, for which we directly take its correction as the translation.

Multiple contexts facilitate generalization on *Unseen Context*. From Table 2, we can see that despite that the Teacher model achieves significantly better performance than the Student model, the SeqKD-Full method can only narrow less than half of the gap. However, by synthesizing more contexts for each error, MT-PATCHER (+PE + PDG) improves the relative performance from 59.2% to 80.5% for chemistry materials, and 57.5% to 69.8% for Chinese Idioms, indicating the importance of translation knowledge in multiple contexts in order to generalize to novel contexts better.

Error Anticipation improves performances on *Unseen Word*. We can also observe that both SeqKD-Full and MT-PATCHER (+PE + PDG) cannot behave well on the *Unseen Word* set, which can be attributed to their inability to extrapolate from the observed errors to unseen errors. By generating analogous words to anticipate more errors, the translation performances on *Unseen Word* are significantly improved, validating the effectiveness of the proposed error anticipation method.

5 Discussion

We provide further analysis on how MT-PATCHER works and its applicability to real-world scenarios. All experiments are conducted on the WMT22 Chinese-to-English translation datasets, and the student MT model is NLLB 3.3B.

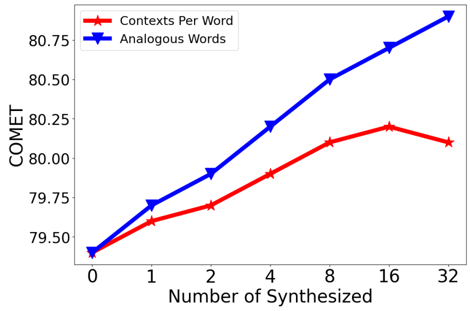

Figure 2: Translation performance as the number of synthesized contexts per word and analogous word grows.

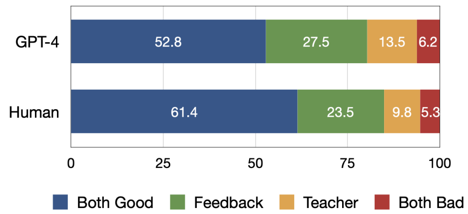

Figure 3: Comparison of translation quality on error words between the Teacher's translation and the feedbacker's correction.

5.1 Impact of the number of synthesized contexts per word and analogous word

In Figure 2, we plot how increasing the number of synthesized contexts per word and analogous words affects the translation performance of the student model. Note that we only synthesize one context for each analogous word. We can see increasing both numbers results in improved translation performance. For synthesized contexts, the gain plateau between 16 to 32 suggests this amount of different contexts is adequate for word or phrase learning. For analogous words, however, we observe the performance grows at a log-linear rate [3].

5.2 Does asking for feedback better elicit LLMs' translation knowledge?

We conduct a head-to-head comparison between two ways to leverage the teacher LLM: ask the teacher to directly provide translation vs. ask MT-PATCHER to give feedback on the student's translation. Specifically, we randomly select 1000 examples and compare the correction provided by MT-PATCHER to the translation provided by the teacher. The comparison is made by both human

[3]It is worth noting that this does not mean MT-PATCHER can improve the translation performance endlessly, since it cannot generate an unlimited amount of valid analogous words. The performance will eventually plateau, although we have not scaled to the number due to the computational limitation.

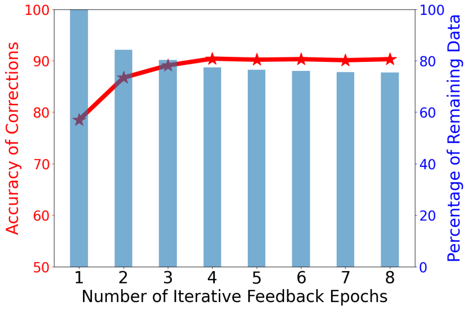

Figure 4: Accuracy of corrections and percentage of remaining data after applying different epochs of iterative feedback.

	COMET	BLEURT	BLEU
$k = 1$	79.4	67.0	24.2
$k = 2$	79.8	67.5	24.7
$k = 3$	80.0	**67.6**	24.9
$k = 4$	**80.1**	**67.6**	**25.1**
$k = 5$	**80.1**	67.5	25.0
$k = 6$	80.0	**67.6**	24.8
$k = 7$	79.8	67.4	24.9
$k = 8$	**80.1**	**67.6**	24.9

Table 4: Translation performance of NLLB-3B model finetuned on post-editing data after k epochs of iterative feedback.

and GPT-4.

The results are shown in Figure 3. It can be seen that MT-PATCHER's corrections are considered by both GPT-4 and human evaluators to be comparable or better than the teacher's translation on more than 80% examples, demonstrating the benefits of eliciting LLM's knowledge in the form of feedback.

5.3 The Effectiveness of Iterative Feedback

In this section, we explore whether the application of iterative feedback on post-edited translations can enhance the final translation quality, thereby yielding a better Student model. While iterative feedback may incur additional computational costs, it allows us to compare feedback across multiple iterations and assess the reliability of error identification and correction from the feedbacker. Intuitively, if an error span identified and rectified in the i-th epoch is still deemed problematic in the subsequent epoch, it suggests an inconsistency in the feedbacker's decision-making process. To prevent the introduction of incorrect knowledge during the knowledge transfer process, examples with such inconsistencies are discarded.

We randomly select 2000 instances of MT-PATCHER's feedback on NLLB-3B's translation results and apply iterative feedback. We then ask GPT-4 to evaluate the feedback quality after each iterative feedback epoch. The results, depicted in Figure 4, indicate that iterative feedback can enhance the accuracy of corrections in remaining examples, converging to 90.4% after 4 epochs at the expense of filtering out approximately 20% of examples. To understand the quality-quantity trade-off of demonstration data, we further fine-tune the Student NLLB model on post-editing data after each iterative feedback epoch and display the translation performance in Table 4. Despite a decrease in the amount of fine-tuning data as the epoch increases, the translation performance of the fine-tuned model continues to improve, highlighting the significance of high-quality fine-tuning data.

	NLLB		ParroT	
	ZH→EN	EN→DE	ZH→EN	EN→DE
Student	76.8	86.1	75.4	80.5
SeqKD-Full	79.5	84.8	76.5	80.9
NLLB†	80.3	87.2	77.5	81.3
ParroT†	79.9	86.8	78.2	81.8

Table 5: Translation performances when applying MT-PATCHER trained on one student model to another. Performances are evaluated by COMET score. Models with † are MT-PATCHER (+ PE + PDS + WA) trained for the corresponding MT model. For reference, we also list the performances of the original student model and SeqKD-Full baselines.

5.4 Transferability of MT-PATCHER

The construction of MT-PATCHER is model-dependent; that is given an MT model, LLMs are finetuned on the data from GPT-4 which demonstrates how to execute the MT-PATCHER pipeline on the translation of the corresponding MT model. Considering the cost of data collection and model training, one may question whether MT-PATCHER is transferable, i.e., a patcher model for one MT model can improve the performance of another MT model. We present such results in Table 5. Although the performance of applying MT-PATCHER to its dedicated MT model is superior, the application of MT-PATCHER trained on another model still significantly surpasses the baseline results, suggesting the potential for a robust MT-PATCHER across various MT models.

6 Conclusion

We introduce MT-PATCHER, a framework designed to leverage capabilities of LLMs to enhance the efficiency and effectiveness of translation knowledge transfer from LLMs to existing MT models. Our approach involves a pipeline that initially generates feedback on translations produced by MT models, followed by the synthesis of potential errors and diverse contexts to systematically rectify these translation errors. Through experimentation on both general and narrow domain MT benchmarks, we demonstrate that MT-PATCHER effectively improves student MT models' performances compared to SeqKD baselines, and exhibits successful transferability across different models.

In the future, we plan to refine our method from two angles. Firstly, previous works (Freitag et al., 2019; Riley et al., 2020) have identified translationese as a significant issue, and training on pseudo data generated by LLMs can exacerbate this problem. A promising solution could involve retrieving target sentences containing correction words and back-translating them to the source side. Secondly, the feedback's *reason* field contains a wealth of valuable information. We intend to explore more efficient strategies to harness this data.

Limitations

Our method focuses on transferring translation knowledge, especially long-tailed lexical knowledge from LLMs to existing MT models, which cannot solve all kinds of translation errors, such as misunderstanding the sentence structure, over/under-translation, etc.

We leverage GPT-4 as evaluators in multiple experiments in this paper. Despite its evaluation has been shown to correlate with human beings well in many previous works, there is still knowledge deficiency in itself and cannot guarantee that the evaluation contains no errors.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang and Shanbo Cheng are the corresponding authors. This work is supported by National Science Foundation of China (No. 62376116, 62176120), the Liaoning Provincial Research Foundation for Basic Research (No. 2022-KF-26-02).

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In Findings of the Association for Computational Linguistics: ACL 2023, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Baichuan Inc. 2023. Baichuan 2: Open Large-scale Language Models.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. CoRR, 2005.14165.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024. Teaching large language models to self-debug. In The Twelfth International Conference on Learning Representations.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 575–593, Toronto, Canada. Association for Computational Linguistics.

Joyce L. Epstein and Frances L. Van Voorhis. 2001. More than minutes: Teachers' roles in designing homework. Educational Psychologist, 36(3):181–193.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at scale and its implications on MT evaluation biases. In Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers), pages 34–44, Florence, Italy. Association for Computational Linguistics.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pages 10421–10430. PMLR.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In International Conference on Learning Representations.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are GPT models at machine translation? a comprehensive evaluation. CoRR, cs.CL/2302.09210v1.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. ArXiv, abs/1503.02531.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Findings of the Association for Computational Linguistics: ACL 2023, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Wenxiang Jiao, Jen-tse Huang, Wenxuan Wang, Zhiwei He, Tian Liang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023a. ParroT: Translating during chat using large language models tuned with human translation and feedback. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 15009–15020, Singapore. Association for Computational Linguistics.

Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, and Zhaopeng Tu. 2023b. Is ChatGPT a good translator? Yes with GPT-4 as the engine. CoRR, cs.CL/2301.08745v3.

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. CoRR, cs.LG/2001.08361v1.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Jackson F Lee Jr and K Wayne Pruitt. 1979. Homework assignments: Classroom games or teaching tools? The Clearing House, 53(1):31–35.

Jiahuan Li, Hao Zhou, Shujian Huang, Shanbo Cheng, and Jiajun Chen. 2023. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions. CoRR, cs.CL/2305.15083.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Min Liu, Yu Bao, Chengqi Zhao, and Shujian Huang. 2023. Selective knowledge distillation for non-autoregressive neural machine translation. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In Thirty-seventh Conference on Neural Information Processing Systems.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. arXiv preprint arXiv:2306.01116.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2685–2702, Online. Association for Computational Linguistics.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7737–7746, Online. Association for Computational Linguistics.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In Proceedings of the 4th Workshop on NLP for

Conversational AI, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.

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.

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2018. Multilingual neural machine translation with knowledge distillation. In International Conference on Learning Representations.

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, 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. 2023. Llama 2: Open foundation and fine-tuned chat models.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting palm for translation: Assessing strategies and performance. CoRR, 2211.09102.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. 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 6456–6466, Online. Association for Computational Linguistics.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging large-scale language models for text augmentation. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias.

Ann Yuan, Daphne Ippolito, Vitaly Nikolaev, Chris Callison-Burch, Andy Coenen, and Sebastian Gehrmann. 2021a. Synthbio: A case study in faster curation of text datasets. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021b. WuDaoCorpora: A super large-scale chinese corpora for pre-training language models. AI Open, 2.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2019. Understanding knowledge distillation in non-autoregressive machine translation. In International Conference on Learning Representations.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. CoRR, cs.CL/2304.04675v2.

	COMET	BLEURT	BLEU
Student	82.4	70.4	26.4
SeqKD-Full	75.9	62.8	22.3
MT-PATCHER	81.7	69.5	26.3

Table 6: Translation performance on WMT22 German → English test set. SeqKD-Full and MT-PATCHER are finetuned student models on pseudo Chinese → English parallel sentences.

Appendix

A Prompts for MT-PATCHER

Table 7, 8, 9, 10 shows the prompt we used for the feedbacker, sentence analysis, parallel data synthesis and word analogy task, respectively.

B Implementation details

We fully finetune LLMs on the collected demonstration data from GPT-4 for 3 epochs. The learning rate is set to $1e-5$, and the batch size is 64. During training, we only compute the next token prediction loss on the response tokens.

C MT-PATCHER suffers less from catastrophic forgetting.

We test the German→English performance of competitors in the Chinese→English setting, including the original student model (ParroT-7B), SeqKD-Full, and MT-PATCHER (PE). We found SeqKD-Full experiences a significant decrease in performance, while MT-Patcher's performance degradation is much less. This suggests that MT-PATCHER is less prone to catastrophic forgetting, thereby demonstrating its potential for repeated application to a target MT system without detriment to its initial capabilities.

D Details of datasets used for chemistry materials and Chinese idioms

For chemistry materials, the data is extracted from *Inventory of Existing Chemical Substances in China*, released by Ministry of Ecology and Environment, China [4].

For Chinese idioms, we use the crawled data from the Github repo [5], and have manually checked the data quality (Of the randomly selected 50 examples, there are only 2 examples that have quality issues).

[4] https://www.mee.gov.cn/gkml/hbb/bgg/201301/t20130131_245810.htm

[5] https://github.com/pwxcoo/chinese-xinhua

We split each word set to two subsets with 5500 and 500 words, respectively, and use GPT-4 to synthesize contexts for them. Figure 6 illustrates the process of constructing the monolingual set and two test sets.

E Prompts for Evaluation

Table 11 shows the prompt we used for evaluating the translation quality of Chinese idioms. Table 12 shows the prompt we used for translation comparison between direct generation and feedback.

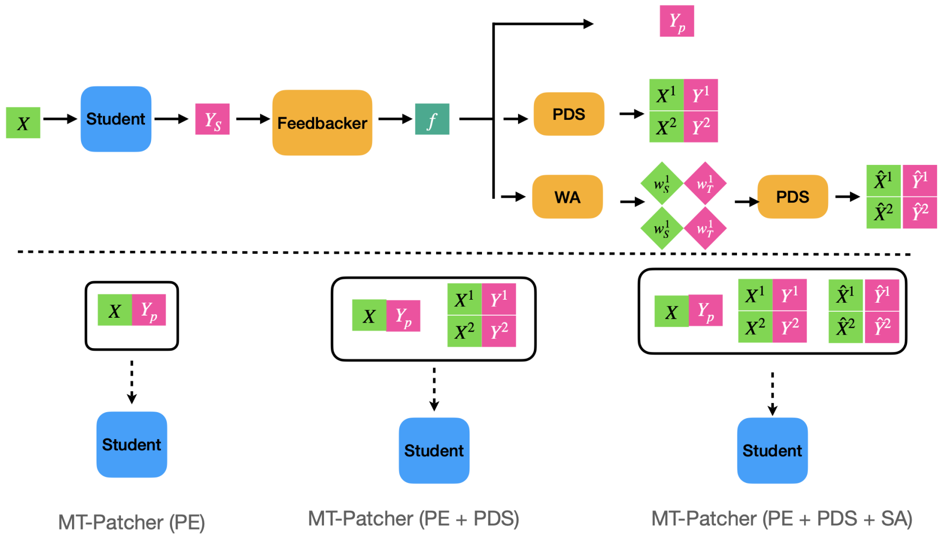

Figure 5: Illustration of variants of MT-PATCHER. PDS denotes the parallel data synthesizer, and WA denotes the word analoger.

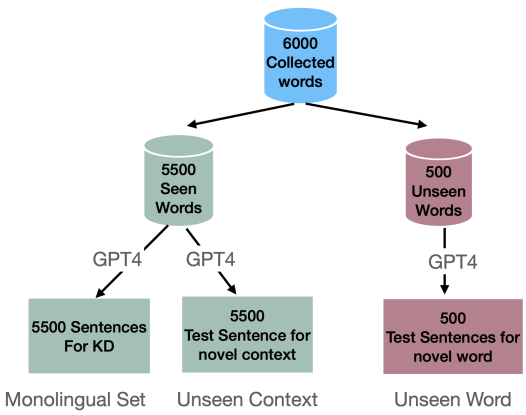

Figure 6: Illustration of the process how the monolingual set and two test sets are splitted from initial collected word sets.

```
 Assuming you are a highly proficient translator skilled at providing detailed and comprehensive
assessments of machine translations. I will give you a <srclang> sentence X and its <tgtlang>
translation Y, and I would like you to help assess the translation.
1. You should first provide an overall assessment.
2. Following that,
- If there are no errors, just say "No error." and do not provide an explanation.
- If there are errors, please specify
- the error type,
- the corresponding segment in the <srclang> sentence X,
- the corresponding segment in the translation Y,
- the reason for the error,
- and the correct translation for the segment
- If there are errors, you should also provide a good translation at the end of the assessment.
4. For multiple errors, you should address them separately.
5. Try to pinpoint the smallest segments containing errors and explain them, avoiding cases where
the error encompasses the entire sentence.
6. Carefully read the original text and the translation to identify all translation errors.
7. Your response should be in English.
8. Be concise.

Now, please assess the following translation:

<srclang>: <srctext>
<tgtlang>: <tgttext>

Assessment:
```

Table 7: Prompt that we use for the feedbacker task.

```
 Suppose you are a language expert of <srclang> and <tgtlang>. Given a sentence X, please point
out its topic, domain and style.
Input:
X: <srctext>
Output:
```

Table 8: Prompt that we use for the sentence analysis task.

```
 Suppose you are a language expert of <srclang> and <tgtlang>. Given a topic, a domain and a
style, as well as a bilingual word pair, please generate a pair of parallel sentences that adhere
to the given topic, domain and style. They should also contain the given word pair.
Input:
Domain: <domain>
Topic: <topic>
Style: <style>
Word Pair: <wordpair>

Output:
```

Table 9: Prompt that we use for the parallel data synthesizer task.

```
Assume you are a <srclang> and <tgtlang> language expert with a wealth of knowledge and associative ability in both languages. I will give you a word/phrase P from an <srclang> sentence X. Please associate from the following aspects and generate three words similar to X for each aspect, and provide the <tgtlang> translation of these words.

Aspects of association:
- Category. What kind of category does this word belong to?
- Semantics. What words often appear in the same context as the given word?

NOTE, the associated words should be rare words, so that it is unlike for a machine translation system to translate it correctly.

Input:
X: <srctext>
P: <errorword>

Output:
```

Table 10: Prompt that we use for the word analogy task.

```
Assume you are a language expert in English and Chinese.  I will give you a Chinese idiom S, a sentence X that contains S, and a machine-generated English translation Y of the source sentence X. I will also give you the explanation/definition E of the idiom S. Your task is to first identify the translation of S in Y, and judge whether the translation of the idiom is correct.

Note:
1. The score range is 0/1/2/3/4/5, where
- 0: Completely incorrect translation or no translation
- 1:  Literal translation of the original, without conveying any implied meaning, leaving non-Chinese background readers baffled
- 2:  Literal translation of the original, partially conveying the implied meaning, easy for non-Chinese background readers to understand
- 3: Interpretative translation of the idiom, but only partially conveying the implied meaning
- 4: Interpretative translation of the idiom, fully conveying the implied meaning
- 5:  The translation perfectly conveys the implied meaning of the idiom, is very easy for all readers to understand, and also maintains the aesthetic sense of the original

2. You should generate the explanation of your decision concisely.
Now, please process the following inputs:
```

Table 11: Prompt that we use for evaluating the quality of translating Chinese idioms.

```
Assume you are a language expert in Chinese and English. I will give you a sentence X, the word P in that sentence, and two translations of the sentence X: A and B. Your task is to assess which translation contains the correct translation of the word P.

Requirements:
(1) Ignore other differences between the two translations. Only compare the translation of the word P.
(2) Your answer should first state the reason for your comparison, and then give your comparison.
(3) Your comparison should be A, B, C and D.
- A: the first translation of the word P is better.
- B: the second translation of the word P is better.
- C: Both are fine.
- D: Both are bad.

Now, please process the following inputs:
```

Table 12: Prompt that we use for comparing translations from direction generation and feedback.