Guiding In-Context Learning of LLMs through Quality Estimation for Machine Translation

Javad Pourmostafa Roshan Sharami
Dimitar Shterionov
Pieter Spronck
Department of Cognitive Science and Artificial Intelligence, Tilburg University

j.pourmostafa@tilburguniversity.edu
d.shterionov@tilburguniversity.edu
p.spronck@tilburguniversity.edu

Abstract

The quality of output from large language models (LLMs), particularly in machine translation (MT), is closely tied to the quality of in-context examples (ICEs) provided along with the query, i.e., the text to translate. The effectiveness of these ICEs is influenced by various factors, such as the domain of the source text, the order in which the ICEs are presented, the number of these examples, and the prompt templates used. Naturally, selecting the most impactful ICEs depends on understanding how these affect the resulting translation quality, which ultimately relies on translation references or human judgment. This paper presents a novel methodology for in-context learning (ICL) that relies on a search algorithm guided by domain-specific quality estimation (QE). Leveraging the XGLM model, our methodology estimates the resulting translation quality without the need for translation references, selecting effective ICEs for MT to maximize translation quality. Our results demonstrate significant improvements over existing ICL methods and higher translation performance compared to fine-tuning a pre-trained language model (PLM), specifically mBART-50.

1 Introduction

Pre-trained large language models (LLMs) quickly gained popularity (and continue to do so) due to their performance on a large set of natural language processing (NLP) tasks, including machine translation (MT) (Zhu et al., 2023; Xu et al., 2024). However, the accuracy of their outputs is significantly influenced by the quality of the *in-context* examples (ICEs) provided to them (Jiang et al., 2020; Alves et al., 2023).[1] If these examples do not align well with the specific task and source domain, the LLMs' outputs can be inaccurate. Therefore, there is a critical need to develop (better) methods for selecting appropriate examples that match the task and source domain being translated. These methods collectively fall under the umbrella of in-context learning (ICL) (Liu et al., 2022).

Traditionally, creating ICEs for MT involves either random selection (Sia and Duh, 2023) or us-ing a strategy such as maximizing an evaluation metric like BLEU, to choose examples that improve the metric (Agrawal et al., 2023). The former was initially used for its simplicity and ease of implementation. However, relying on randomness can lead to inconsistent results and pose significant computational costs (Lu et al., 2022). Recent state-of-the-art (SOTA) ICL approaches focus on retrieving training examples that are closely relevant to the context of source sentences of test sets using unsupervised retrievers, such as BM25 (Robertson and Zaragoza, 2009). Recent studies have also shown that a range of factors, such as order (Lu et al., 2022), template (Jiang et al., 2020), domain, and number of ICEs, significantly impact the performance (Agrawal et al., 2023; Raunak et al., 2023a).

Naturally, the most effective ICEs for a given source text are the ones that would impact the resulting translation quality, which would ultimately de-

---

[1] For simplicity, we sometimes refer to it as "example(s)" throughout this paper.

pend on translation references or human judgment. In MT, quality estimation (QE) has become a standard approach for evaluating an MT system's output without relying on reference translations Blain et al. (2023). Recently, Lee (2020), Ye and Li (2023), and Sharami et al. (2023) showed the effectiveness of domain-specific QE when it comes to domain-specific MT (in contrast to the ineffectiveness of generic QE). Building on this and to address the aforementioned challenges, our work proposes to leverage domain-specific QE to assist in the selection of ICEs, with the goal of determining the suboptimal number and combination of ICEs to maximize MT quality, all without reference translations. As QE would assess the impact of different ICE combinations and sequences, we hypothesize that this integration has the potential to not only improve translation performance but also reduce processing time, as QE could result in smaller sets of ICEs, which would reduce the inference times (Petrov et al., 2023). This is particularly crucial considering the limited number of ICEs that can be fed into LLMs (Agrawal et al., 2023). Therefore, our study aims to investigate the feasibility of selecting ICEs on a per-source basis. Specifically, we aim to answer the following research question (RQ): *How effective are domain-specific QE models in determining ICEs for translation tasks in an LLM?*

Our proposed ICL methodology for MT combines an unsupervised retriever to select ICEs with QE to assess their impact on the translation quality, determining which ICE combination to include. Instead of feeding all selected examples, we only select examples whose QE points to maximizing the LLM translation quality.

Our findings on German-English translations demonstrate that our proposed approach outperforms the current SOTA ICL methods for MT as well as a fine-tuned mBART-50 (Tang et al., 2020).

2 ICL Using Quality Estimation for MT

To utilize LLMs for effective MT, as noted in Section 1, what is needed is a set of examples to provide the context (and thus guide or steer the LLM toward a correct, context-specific translation) — that is, a set of ICEs — and what is further important is the number of ICEs and their combination.[2] Ultimately, what is required is that the ICEs provide

context that is neither too specific nor too broad and can effectively boost the translation. Our goal with this work is to develop a methodology that optimizes both these aspects in order to deliver high-quality MT. Our methodology for identifying effective ICEs involves two key components: (1) an unsupervised retriever that locates examples closely related to the sentence to be translated and (2) a search algorithm that uses QE to select a combination of examples that leads to the improvement of translation quality, i.e., aiming to maximize the BLEU score.

2.1 Unsupervised Retriever Ranking

We employ the *BM25* ranking algorithm (Trotman et al., 2014) due to the effective utilization of unsupervised retriever methods demonstrated in previous research, such as (Agrawal et al., 2023). BM25 sorts training pairs (source text and their translations) based on their relevance to a given query, i.e. the sentence to be translated. Subsequently, we select the top K sentence pairs ranked by the algorithm, where K is a hyperparameter that controls the number of pairs to be fed into the search algorithm.

2.2 Search Algorithm Coupled with QE

Our search algorithm comprises three main phases: *Selection, Translation*, and *Estimation*. During the Selection phase, the algorithm selects the highest-ranked training example from the initial ICEs provided by the unsupervised retriever ranking method (out of K ICEs). This selected example is then concatenated with the previously selected ICEs. In the first iteration, no ICEs have been selected before. In the Translation phase, the selected ICE is translated by the model. In the Estimation phase, the LLM output (translated text) and the original source text are inputted into the domain-specific QE model to estimate the quality of the translation. Our proposed methodology relies on sentence-level QE.

Next, the selected ICE, together with its estimated quality and the LLM translation output, are appended to an intermediate list. To track the highest quality obtained thus far, the algorithm sorts the list in descending order based on the estimated quality. To avoid duplication, the selected ICE is removed before the next iteration. This iterative process continues until the best-estimated translation quality no longer improves within the specified pa-

---

[2]The question of the order of examples is not specifically discussed in this paper but is left for future work.

tience threshold. Alternatively, the process terminates once all K ICEs have been selected.

This methodology allows for the systematic selection of ICEs that improve translation quality compared to previous ICL methodologies while efficiently managing the computational resources required for the search process. This efficiency is achieved by integrating early stopping conditions with predetermined patience. Notably, we do not explore permutations of initial ICEs, as doing so would require a large number of attempts, leading to high computational costs during the search process.[3]

3 Experiments Setup

We conducted four main experiments to test the effectiveness of our methodology. Three of these experiments compare our methodology to existing ICL ones in different settings, or *Modes*. The fourth experiment compares our methodology to a fine-tuned mBART-50, aiming to assess which method is preferred (with respect to obtaining better translations).

It is important to note that we do not fine-tune the LLM. The process of building the QE model used in our experiments is detailed in Section 3.2.

3.1 Search Algorithm

We conducted experiments using the search algorithm outlined in Section 2.2 across three operational modes:

Mode 1: This mode uses QE with ICEs ordered by BM25 to assess the effectiveness of combining BM25 and QE in the proposed ICL methodology.

Mode 2: This mode investigates the impact of ordering ICEs by n-gram overlap, particularly unigrams, alongside QE, on the proposed methodology. Given the success of ordering ICEs based on their n-gram overlap match with the source, as demonstrated in (Agrawal et al., 2023), we assess how this ordering, based on ICEs' n-gram overlap with the source text, influences the translation quality. This involves reordering ICEs according to their n-gram overlap, which is calculated using the NLTK word tokenizer. Higher overlap matches prioritize ICEs in the list and feed them into LLMs earlier.

Mode 3: Instead of relying on QE, in this mode, we compute the BLUE score on the existing **test set**. This approach is not a realistic case, but it is the most favorable scenario, and we use it as the highest bound to compare with Mode 1.

The search algorithm generates up to 16 candidates. In each mode, we conducted experiments using three early stopping patience values (3, 8, and 16), determining the maximum number of ICEs (K) generated. We included Patience 16, which implies no early stopping, to evaluate the model's performance with the maximum ICEs. Additionally, the search process halts if the estimated label reaches or exceeds 100, preventing further evaluations.

3.2 Quality Estimation

Following (Ranasinghe et al., 2020; Lee, 2020; Sharami et al., 2023), we develop a domain-specific QE model. First, we trained a QE model using out-of-domain (OOD) data (as detailed in Section 3.2.1) to ensure generalizability; and second, we fine-tuned the model using the training set described in Section 3.4 to provide domain-specific QE model and address domain mismatch, which is critical (Koehn and Knowles, 2017).

In our experiments, we used BLEU as the quality label because our study focused on translation performance rather than post-editing effort, which is typically evaluated using (H)TER (Specia and Farzindar, 2010). We employed the "MonoTransQuest" architecture from the TransQuest framework (Ranasinghe et al., 2020), known for its success in prior QE studies. However, instead of employing softmax computation, we directly utilized logits to estimate the quality labels. This strategy saves computation time, as softmax computation can be resource-intensive (Ruder, 2016).

3.2.1 QE data

We utilized the German-English "EuroPat" dataset, accessed through Opus (Tiedemann, 2012), to develop our generic QE model. We chose this dataset because it provides ample data samples, ensuring broad coverage of vocabulary — a critical aspect in developing generic models.

However, as MT datasets like EuroPat typically consist of pairs of source and translated text, it was necessary to synthetically create post-editing text

---

[3]A pseudocode outlining the search methodology can be found in Algorithm 1 in the Appendix. The phases of translating a source text of a test set using our methodology are depicted in Figure 2.

(since the QE data creation process requires a triplet input: source text, machine-translated text, and post-edited text). To accomplish this, we used a pre-trained multilingual MT model, namely mBART-50 that supported the language pair used in our experiment. This involves translating 1M randomly chosen source texts from EuroPat. Afterward, the resulting translations were considered as machine-translated text, with the corresponding reference translations acting as post-edited text.

Using SacreBLEU, we calculated the BLEU score, comparing the translated text with its corresponding post-edited text. This approach, which has been demonstrated to be effective in QE (Negri et al., 2018; Lee, 2020; Sharami et al., 2023), enabled us to use the source and (machine-) translated text as input and the BLEU score as the target value for the QE model. For building domain-specific QE, we utilized the training set detailed in Section 3.4 and applied the aforementioned approach to synthetically generate BLEU scores for the entire dataset.

3.3 Multilingual Large Language Model

For our experiments and hypothesis validation, we used XGLM (Lin et al., 2022). This choice stems from the outstanding performance of the model in the MT field. This also ensures a fair comparison of our proposed methodology with previous research, such as (Agrawal et al., 2023), which introduced SOTA approaches in ICL for MT.

We used the 7.5 billion-parameter XGLM implementation and tokenizer by Hugging Face[4], consistent with previous research. We employed a template from Lin et al. (2022) to maximize translation performance. $</s>$ serves as the ICE separator in this template. "BLANK" denotes an empty string within the template.

$$\{\text{source text}_1\} = \{\text{target text}_1\} </s>$$
$$\{\text{source text}_2\} = \{\text{target text}_2\} </s>$$
$$\ldots = \ldots </s>$$
$$\{\text{source text}_n\} = \text{BLANK}$$

3.4 Dataset and Evaluation Metrics

We used a dataset comprising German-to-English translation pairs within the IT domain, sourced from (Aharoni and Goldberg, 2020). This dataset

was chosen due to the challenges that MT systems and LLMs face when translating out-of-domain contexts, particularly in specialized fields, as noted in previous studies (Koehn and Knowles, 2017; Agrawal et al., 2023). The specialized and constrained nature of the IT domain provided an ideal setting for evaluating our methodology's performance under these conditions.

The dataset utilized in this study consisted of approximately 222k training sentences, 2k development sentences, and 2k test sentences. To assess the translation effectiveness of the models, we employed metrics such as BLEU from Sacre-BLEU (Post, 2018) and COMET (Rei et al., 2020).

3.5 Number of ICEs

We use between 1 and 16 ICEs. These may originate either from a random approach or from an advanced (guided) selection. To keep these separated in our analysis, we designate two different counts – p and q. This choice and naming convention is grounded by previous research exploring the impact of varying ICE numbers. While our study explicitly caps the upper limit of q at 16, values spanning from 1 to 16 remain feasible options — unlike the fixed value in the compared systems.

3.6 Compared Systems

We conducted a comparative analysis with methods from previous studies; *random* and *task-level sampling*, *BM25*, *R-BM25*, and *fine-tuned mBART-50*.

Random: We conducted three random trials, generating random numbers based on parameter p. These numbers, ranging from 1 to the size of the training set, selected corresponding translation pairs. To create the prompt[5], in addition to the training examples (i.e., ICEs), we need the source side intended for translation. We utilize the source from the development set, in contrast to the advanced methods in ICL, where the source text from the test set is typically employed. The reason for selecting the development set over the test set in this approach is that development sets are generally from the same distribution, domain, and context as the test set. This similarity increases the likelihood that the examples in the development set will better match the content and context of the test set, thereby enhancing the rel-

---

[4] https://huggingface.co/docs/transformers/model_doc/xglm

[5] In the literature, the term "prompt" is frequently used interchangeably with "ICE"

evance and effectiveness of the prompts.

The generated prompt is inputted into the LLM for translation. Then, the BLEU score of the development set is computed. The random number that produces the highest score among the trials is selected, and the training examples linked to this number are concatenated with the test set's source text.

Task-level: Based on the work of Agrawal et al. (2023), the task-level approach is similar to the random approach but differs in the number of trials used. We employ 100 trials for the task-level approach, a significantly higher number than the random approach. The reason for using more trials is to generate a greater variety of ICEs, aiming to enhance the performance of LLMs in the translation task. However, this results in longer execution times compared to the random approach.

BM25: Using the Moses Tokenizer (Koehn et al., 2007), we first tokenize the training set's source samples. Then, a BM25 model is created for the tokenized corpus by employing the *BM25Okapi* implementation within the *rank_bm25* package.[6]

Next, the test set is tokenized using the tokenized source. The algorithm then searches for similar training samples based on BM25 criteria, selecting the top q matches for the model. This methodology utilizes the test set as opposed to random and task-level approaches using the development set.

Re-rank BM25 (R-BM25): BM25 aims to find translation examples with the highest n-gram overlap with the source sentence (Luo et al., 2023). However, since retrieved examples score independently, top matches may lack coverage of all source n-grams. This poses an issue in ICL due to LLM input size limitations. To address this, Agrawal et al. (2023) proposed R-BM25. R-BM25 employs a recall-based n-gram overlap (Agrawal et al., 2023) to extract word n-grams and their numbers from the test source and BM25 retrieved examples.

Fine-tuning mBART-50: Different ICL methodologies, including our own, are assessed in comparison to the process of fine-tuning a pre-trained multilingual MT model, specifically mBART-50. The selection of mBART-50 is based on its alignment with the language specifications of the experiment

and its proven track record of achieving success in MT tasks through the utilization of pre-trained models (Yuan et al., 2022; Pham et al., 2022). The fine-tuning of mBART-50 is carried out using the training data outlined in Section 3.4.

3.7 Computational Costs

We monitored and reported the computational costs of the models utilized in our experiments using the *carbontracker* package.[7] This involved calculating the carbon footprint (CO_2eq) emissions, time to prediction (TTP), and electricity consumption (kWh) associated with our experiments. Our experiments were conducted using NVIDIA A40 GPUs.

The script for running our experiments is publicly available at https://github.com/JoyeBright/ICLviaQE/.

4 Experiments Results

This section presents the results of our experiments. To ensure a fair comparison, we conducted a statistical analysis test (t-test) to determine if our models significantly outperformed the baseline.

Comparing to previous work, the results shown in Table 1, indicate that R-BM25 with 16 ICEs outperforms other methods. It is notable that there is a positive correlation between the number of examples and evaluation scores (consistent through all methods – Random, Task-level, BM25, and R-BM25), although at the expense of prediction time (i.e., TTP). Employing 16 examples significantly improved performance compared to using only one example in the random approach.

Analyzing the performance of our methods in Mode 1 (referred to as "M 1", with P = 3, 8, or 16 in Table 1), we observe that our proposed methodology with different patience thresholds consistently outperforms all previous methods, including the baseline. This trend holds for both the COMET and BLEU metrics across all the methods. Specifically, our method exhibits a minimum improvement of 0.52 points in the BLEU score (from 45.20 to 45.72) with patience threshold of 3 and a maximum improvement of 1.58 points in the BLEU score (from 45.20 to 46.78) with a patience threshold of 16 compared to R-BM25 with 16 examples.

---

[6]https://github.com/dorianbrown/rank_bm25
[7]https://github.com/lfwa/carbontracker

Method	$p + q$	BLEU	COMET	TTP (hh:mm)	CO2 (kg)	GPU (kWh)
Random	1 + 0	10.38	0.6895	01:51	00.13	00.39
Random	16 + 0	31.65	0.7844	02:20	00.19	00.58
Task-level	1 + 0	29.17	0.7586	62:50	09.83	29.10
Task-level	16 + 0	32.88	0.8083	78:30	12.80	35.91
BM25	0 + 1	39.24	0.7833	00:56	00.06	00.19
BM25	0 + 16	44.50	0.8120	00:58	00.07	00.19
R-BM25	0 + 1	40.88	0.7990	01:01	00.06	00.21
R-BM25	0 + 16	**45.20**	**0.8218**	01:04	00.07	00.21
M 1, P = 3	0 + 16	45.72	0.8395	01:49	00.22	00.67
M 1, P = 8	0 + 16	**46.43**	**0.8501**	03:48	00.50	01.51
M 1, P = 16	0 + 16	**46.78**	**0.8554**	05:11	00.68	02.05
M 2, P = 3	0 + 16	46.05	0.8400	01:30	00.21	00.64
M 2, P = 8	0 + 16	**46.59**	**0.8518**	03:52	00.51	01.52
M 2, P = 16	0 + 16	**46.52**	**0.8564**	05:00	00.66	02.01
M 3, P = 3	0 + 16	49.89	0.8532	01:36	00.22	00.66
M 3, P = 8	0 + 16	**52.63**	**0.8725**	03:14	00.45	01.40
M 3, P = 16	0 + 16	**53.50**	**0.8791**	04:08	00.55	01.65
mBART-50	N/A	42.76	0.8659	11:20	01.88	04.82

Table 1: **Method Performance in BLEU and COMET Scores.** M 1 to 3 denotes Mode 1 to 3; P is the patience value; p and q are as defined in Section 3.5. "N/A" (not applicable) indicates that fine-tuning does not use ICEs. Bold font represents the highest translation performance. Two numbers are in bold if they are statistically similar (t-test, $p\_value = 0.05$).

Consequently, our methods in Mode 1 are ranked based on their performance, with patience 3 being the least effective model, followed by patience 8, and finally patience 16, representing the most effective method. This ranking indicates that increasing the patience threshold can significantly enhance the translation performance. However, the improvement with patience 16 is not statistically significant compared to patience 8, suggesting that more ICEs do not necessarily enhance translation performance. Similarly, while more substantial contextual improvement (as indicated by the COMET) is observed at the maximum patience threshold (16), it is not statistically significant compared to patience 8.

The Mode 2 results demonstrate that all three patience thresholds surpass the methods in the literature. However, this improvement is not statistically significant when compared with the respective experiments in Mode 1. This suggests that ordering the examples according to n-gram (unigram) simi-

larity does not enhance the translation performance in our methodology.

When it comes to Mode 3, we should stress that this is an unrealistic scenario, but used as the highest bound. The results indicate that with a patience of 3, the BLEU score is 4.17 points lower (49.89-45.72). With a patience of 8, this gap increases to 6.2 points (52.63-46.43), and with a patience of 16, it widens further to 6.72 points (53.50-46.78). These differences arise from the QE model estimations in our experiment compared to the scenario where reference labels are available to the search algorithm.

4.1 Time to Prediction (TTP)

Among the methods examined, task-level execution required the most time, with approximately 62 hours for one example and 78 hours for 16 examples. Our method (Mode 1) with a patience value of 16 is relatively time-intensive, taking approximately 5 hours, while a patience value of 3 is comparable to the baseline method, differing by only around 50 minutes. Mode 2 is nearly equivalent to Mode 1 in terms of TTP, whereas Mode 3, where the reference labels are accessed, requires less time than Modes 1 and 2. In addition, the search algorithm incorporates a termination condition, and given that QE estimation rarely triggers this condition, numerous ICEs are left unattempted, resulting in significant time savings.

It is also important to note the time required to train the QE models used in the prediction process. As provided in Appendix 3, the training time for the generic QE model is $+/-$ 5 hours and 55 minutes, while the specific QE model takes about $+/-$ 6 hours and 54 minutes. Although these training times are significant, it is crucial to recognize that QE models, similar to MT models, can be reused for the same language pair and domain, thereby amortizing the initial training cost over multiple predictions.

The last row of Table 1 shows the scores of the translations obtained with the mBART-50 model fine-tuned on the same training set as in ICL. Despite mBART-50 being tailored for MT across 50 languages, it did not outperform the R-BM25 method with 16 examples (best from the existing methods); it was better only than Random, Task-level, BM25, and R-BM25, each with only 1 example. However, when considering translation performance from a contextual perspective, the COMET results indicate that fine-tuning mBART-50 leads to

superior performance compared with lexical overlap. Nevertheless, fine-tuning took significantly longer than identifying ICEs and obtaining inferences from the XGLM.

Compared to our methodology, especially when considering the least performing method (M 1, P = 3), it is significantly worse – 6.47% (42.76 to 45.72). This highlights the substantial efficacy of ICL compared to fine-tuning. Nonetheless, it is noteworthy that various factors might contribute to this observation: e.g., the model's size might be a critical factor, especially during deployment, where larger models like XGLM could pose challenges.

5 Analysis

Output analysis Pre-trained LLMs often exhibit over-generation, i.e., the generation of a larger number of tokens than expected by a human (in comparison to a reference), necessitating extensive post-processing (e.g., post-editing) (Bawden and Yvon, 2023). Figure 1 shows the tokenized output lengths (translations) for our model (Mode 1, patience 8),[8] alongside the R-BM25 with 16 examples. The analysis shows that the length distributions for both models align with the reference distribution, suggesting that the models do not over-generate.

To quantitatively compare these distributions to the reference, we employed the Kolmogorov-Smirnov (KS) test (Kolmogorov, 1933). The results indicate that for R-BM25 versus the reference, the KS statistic is relatively high (0.0749), reflecting a significant difference between the translation lengths of R-BM25 and the reference distribution. The extremely low p-value (2.39×10^{-5}) further confirms this significant discrepancy. Conversely, for Mode 1 with P=8 versus the reference, the KS statistic is considerably lower (0.0232), indicating a much smaller difference in translation lengths. The higher p-value (0.6451) suggests no significant difference, implying that the distribution of Mode 1, P=8 is similar to the reference distribution.

These findings suggest that our proposed methodology could yield translations closer in length to the reference, potentially reducing the need for labor-intensive post-processing efforts and enhancing computational efficiency.

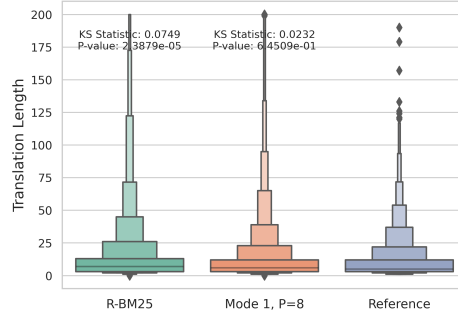[8]Our other models in Mode 1 exhibited similar distributions.

Figure 1: **Tokenized Translation Lengths** comparison between R-BM25, our Mode 1, P=8, and the reference. "KS" denotes the Kolmogorov-Smirnov test, with the p-value indicating significance.

ICE Number Analysis The number of selected ICEs holds a significant importance within the ICL algorithm, as it directly impacts the token processing time and the capacity of LLMs to handle additional ICE instances. We analyzed the number of ICEs that our algorithm selected across all three modes. The results (Table 2) show that the minimum number of ICEs selected is 1, while the maximum is: 12 for Mode 1, 16 for Mode 2, and 16 for Mode 3. The average (mean) number of ICEs is found to be lowest in Mode 3 and highest in Mode 1. In addition, Mode 2 results in a reduction in the number of ICEs within our proposed algorithm. The notably lower average number of ICE instances in Mode 3 can be attributed to its access to the test set, allowing for the selection of optimal ICE combinations based on test set performance and activating an early stopping condition if the score exceeds 100. Contrarily, while Mode 1 exhibits similarities to Mode 3, its relatively higher average can be linked to inaccuracies in QE estimation. Moreover, our analysis shows that QE estimations rarely reach a score of 100, thus rendering the early stopping condition inactive.

Mode	Min	Mean	Max
#1	[1, 1, 1]	[2.25, 3.76, 4.84]	[12, 16, 16]
#2	[1, 1, 1]	[2.20, 3.70, 4.74]	[12, 16, 16]
#3	[1, 1, 1]	[2.15, 3.47, 4.47]	[12, 16, 16]

Table 2: **Number of ICEs selected for each mode at different patience thresholds.** Labels [x, y, z] correspond to patience values 3, 8, and 16.

CO$_2$ Emissions Our analysis reveals that using XGLM for translation yields lower CO$_2$ emissions than fine-tuning mBART-50, making it a more environmentally sustainable choice. In Mode 1 of our proposed methodology, with patience 16, XGLM emitted 0.68 KG of CO$_2$, while fine-tuning mBART-50 emitted 1.88 KG. Interestingly, the task-level method with 16 ICEs emitted the highest amount of CO$_2$, totaling 12.80 KG. Our proposed approach leads to higher CO$_2$ emissions than R-BM25.

6 Related Work

ICL for MT. ICL[9] represents a relatively new paradigm in natural language understanding. Unlike traditional fine-tuning approaches, where a PLM undergoes parameter updates using a specific dataset, ICL typically directly generates the output without any modification to its parameters (Radford et al., 2019; Brown et al., 2020). This is achieved by solely providing the model with a few examples, known as ICEs, which prime the PLM to enhance its performance for the given task (Jiang et al., 2020).

As shown by Vilar et al. (2023), the quality of translation is directly proportionate to the quality of ICEs, where quality refers to ICEs being relevant, clear, accurate, and domain-specific. However, considering all ICEs during processing is computationally demanding (Alves et al., 2023). Hence, it is crucial to selectively choose ICEs that can enhance MT quality. Goyal et al. (2022) conducted a study where ICEs were randomly selected. Despite finding that this random selection of ICEs resulted in good translation performance, the neglect of their order, which was identified as important (Liu et al., 2022; Lu et al., 2022), was a drawback in this approach. To address this, methodologies such as (Agrawal et al., 2023) introduced a re-ranking technique (R-BM25). However, their methodology relies solely on n-grams to order examples, which can enhance fluency but may overlook contextual factors. In our approach, we investigated the unigram order of initial ICEs provided by the BM25 algorithm. We leave the in-depth analysis of ICE order for future work. Additionally, Kumar et al. (2023) highlighted the advantages of using multiple features in ICE selection to improve translation quality, while our QE-based approach simplifies ICE selection without needing to generate additional

---

[9]Also referred to as the prompt retrieval method

features, ensuring efficiency.

QE in MT Evaluation. QE models offer a quick solution to the assessment of the overall usefulness of translated text. These models do not rely on reference translations, thereby reducing the human effort required for quality evaluation (Tamchyna, 2021; Murgolo et al., 2022; Zerva et al., 2022; Blain et al., 2023). Similar to MT models, previous studies highlight the importance of domain-specific QE for accurately estimating translation quality across diverse domains (Lee, 2020; Sharami et al., 2023). This is why, in our work, we employed a domain-specific QE model instead of a generic one to enhance the selection of ICEs.

Integrating QE into ICL offers significant, yet largely unexplored, potential. QE can also better capture out-of-domain gender and word-sense-disambiguation errors (Dinh and Niehues, 2023). Additionally, integrating QE can mitigate reference bias, a significant challenge in accurately estimating the output quality of LLMs (Goyal et al., 2023; Raunak et al., 2023b). The introduction of COMET-QE (Raunak et al., 2023a) exemplifies this pursuit, providing a metric tailored to evaluate the quality of perturbed prompts provided to GPT-3 (Brown et al., 2020), aiming to mitigate reference bias. While in our approach, we employ domain-specific QE to guide the selection of ICEs, this underscores the potential of QE in refining LLM inputs (i.e., ICEs).

7 Conclusion

We propose a novel in-context learning (ICL) methodology for enhancing the translation capabilities of large language models (LLMs) while optimizing computational resources. Our approach leverages domain-specific quality estimation (QE) to guide in-context selection, particularly focusing on determining the suboptimal number and the combinations of in-context examples (ICEs). This novel strategy moves beyond the conventional reliance solely on translation references from development sets seen in prior methods.

We evaluated our approach across different modes and early stopping patience values on the German-to-English IT dataset. Our experiments consistently showed the superior performance of our methodology, surpassing all prior works across both

BLEU and COMET metrics. Our method consistently improves BLEU scores, although this comes at the cost of increased computation time. We also investigated the impact of ordering the ICEs based on their unigram overlap with the source text and found it to be not statistically significant. Furthermore, our experiments highlighted the value of ICL compared to fine-tuning a pre-trained large model, namely mBART-50. We also highlighted that our method leads to less carbon emissions while achieving better translation performance.

In the future, we would like to conduct further research on the impact of our proposed methodology across different language pairs, domains and LLMs. Also, we aim to explore alternative metrics beyond BLEU to tailor the selection process, as well as additional features such as bigram, type/token ratio, and length when ordering examples prior to their input into LLMs.

8 Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments.

References

Agrawal, S., Zhou, C., Lewis, M., Zettlemoyer, L., and Ghazvininejad, M. (2023). In-context examples selection for machine translation. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Aharoni, R. and Goldberg, Y. (2020). Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Alves, D., Guerreiro, N., Alves, J., Pombal, J., Rei, R., de Souza, J., Colombo, P., and Martins, A. (2023). Steering large language models for machine translation with finetuning and in-context learning. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11127–11148, Singapore. Association for Computational Linguistics.

Bawden, R. and Yvon, F. (2023). Investigating the translation performance of a large multilingual language model: the case of BLOOM. In Nurminen, M., Brenner, J., Koponen, M., Latomaa, S., Mikhailov, M., Schierl, F., Ranasinghe, T., Vanmassenhove, E., Vidal, S. A., Aranberri, N., Nunziatini, M., Escartín, C. P., Forcada, M., Popovic, M., Scarton, C., and Moniz, H., editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 157–170, Tampere, Finland. European Association for Machine Translation.

Blain, F., Zerva, C., Rei, R., Guerreiro, N. M., Kanojia, D., C. de Souza, J. G., Silva, B., Vaz, T., Jingxuan, Y., Azadi, F., Orasan, C., and Martins, A. (2023). Findings of the WMT 2023 shared task on quality estimation. In Koehn, P., Haddow, B., Kocmi, T., and Monz, C., editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 629–653, Singapore. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Dinh, T. A. and Niehues, J. (2023). Perturbation-based QE: An explainable, unsupervised word-level quality estimation method for blackbox machine translation. In Utiyama, M. and Wang, R., editors, *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 59–71, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., and Fan, A. (2022). The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Goyal, T., Li, J. J., and Durrett, G. (2023). News summarization and evaluation in the era of gpt-3.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020). How Can We Know What Language Models Know? *Trans-

actions of the Association for Computational Linguistics, 8:423–438.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In Ananiadou, S., editor, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In Luong, T., Birch, A., Neubig, G., and Finch, A., editors, *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Kolmogorov, A. N. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83–91.

Kumar, A., Puduppully, R., Dabre, R., and Kunchukuttan, A. (2023). CTQScorer: Combining multiple features for in-context example selection for machine translation. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7736–7752, Singapore. Association for Computational Linguistics.

Lee, D. (2020). Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In Barrault, L., Bojar, O., Bougares, F., Chatterjee, R., Costa-jussà, M. R., Federmann, C., Fishel, M., Fraser, A., Graham, Y., Guzman, P., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Martins, A., Morishita, M., Monz, C., Nagata, M., Nakazawa, T., and Negri, M., editors, *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028, Online. Association for Computational Linguistics.

Lin, X. V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., Pasunuru, R., Shleifer, S., Koura, P. S., Chaudhary, V., O'Horo, B., Wang, J., Zettlemoyer, L., Kozareva, Z., Diab, M., Stoyanov, V., and Li, X. (2022). Few-shot learning with multilingual generative language models. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. (2022). What makes good in-context examples for GPT-3? In Agirre, E., Apidianaki, M., and Vulić, I., editors, *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Luo, M., Xu, X., Dai, Z., Pasupat, P., Kazemi, M., Baral, C., Imbrasaite, V., and Zhao, V. Y. (2023). Dr.icl: Demonstration-retrieved in-context learning.

Murgolo, E., Sharami, J. P. R., and Shterionov, D. (2022). A quality estimation and quality evaluation tool for the translation industry. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 307–308, Ghent, Belgium. European Association for Machine Translation.

Negri, M., Turchi, M., Chatterjee, R., and Bertoldi, N. (2018). ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Petrov, A., Malfa, E. L., Torr, P. H. S., and Bibi, A. (2023). Language model tokenizers introduce unfairness between languages.

Pham, N.-Q., Nguyen, T. N., Nguyen, T.-B., Liu, D., Mullov, C., Niehues, J., and Waibel, A. (2022). Effective combination of pretrained models - KIT@IWSLT2022. In Salesky, E., Federico, M., and Costa-jussà, M., editors, *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 190–197, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

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

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Ranasinghe, T., Orasan, C., and Mitkov, R. (2020). TransQuest: Translation quality estimation with cross-lingual transformers. In Scott, D., Bel, N., and Zong, C., editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Raunak, V., Menezes, A., and Awadalla, H. (2023a). Dissecting in-context learning of translations in GPT-3. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 866–872, Singapore. Association for Computational Linguistics.

Raunak, V., Menezes, A., Post, M., and Hassan, H. (2023b). Do GPTs produce less literal translations? In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1041–1050, Toronto, Canada. Association for Computational Linguistics.

Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. (2020). COMET: A neural framework for MT evaluation. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ruder, S. (2016). On word embeddings - Part 2: Approximating the Softmax. http://ruder.io/word-embeddings-softmax.

Sharami, J. P. R., Shterionov, D., Blain, F., Vanmassenhove, E., Sisto, M. D., Emmery, C., and Spronck, P. (2023). Tailoring domain adaptation for machine translation quality estimation. In Nurminen, M., Brenner, J.,

Koponen, M., Latomaa, S., Mikhailov, M., Schierl, F., Ranasinghe, T., Vanmassenhove, E., Vidal, S. A., Aranberri, N., Nunziatini, M., Escartín, C. P., Forcada, M., Popovic, M., Scarton, C., and Moniz, H., editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 9–20, Tampere, Finland. European Association for Machine Translation.

Sia, S. and Duh, K. (2023). In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. In Utiyama, M. and Wang, R., editors, *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 173–185, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Specia, L. and Farzindar, A. (2010). Estimating machine translation post-editing effort with HTER. In Zhechev, V., editor, *Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 33–43, Denver, Colorado, USA. Association for Machine Translation in the Americas.

Tamchyna, A. (2021). Deploying MT quality estimation on a large scale: Lessons learned and open questions. In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, pages 291–305, Virtual. Association for Machine Translation in the Americas.

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and fine-tuning.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Trotman, A., Puurula, A., and Burgess, B. (2014). Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, page 58–65, New York, NY, USA. Association for Computing Machinery.

Vilar, D., Freitag, M., Cherry, C., Luo, J., Ratnakar, V., and Foster, G. (2023). Prompting PaLM for translation: Assessing strategies and performance. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Xu, H., Kim, Y. J., Sharaf, A., and Awadalla, H. H. (2024). A paradigm shift in machine translation: Boosting translation performance of large language models.

Ye, N. and Li, J. (2023). A k-nearest neighbor approach for domain-specific translation quality estimation. In Feng, Y. and Feng, C., editors, *Machine Translation*, pages 69–80, Singapore. Springer Nature Singapore.

Yuan, B., Li, Y., Chen, K., Lu, H., Yang, M., and Cao, H. (2022). An improved multi-task approach to pretrained model based mt quality estimation. In *CCMT*.

Zerva, C., Blain, F., Rei, R., Lertvittayakumjorn, P., C. De Souza, J. G., Eger, S., Kanojia, D., Alves, D., Orăsan, C., Fomicheva, M., Martins, A. F. T., and Specia, L. (2022). Findings of the WMT 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 69–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Zhu, W., Liu, H., Dong, Q., Xu, J., Huang, S., Kong, L., Chen, J., and Li, L. (2023). Multilingual machine translation with large language models: Empirical results and analysis.
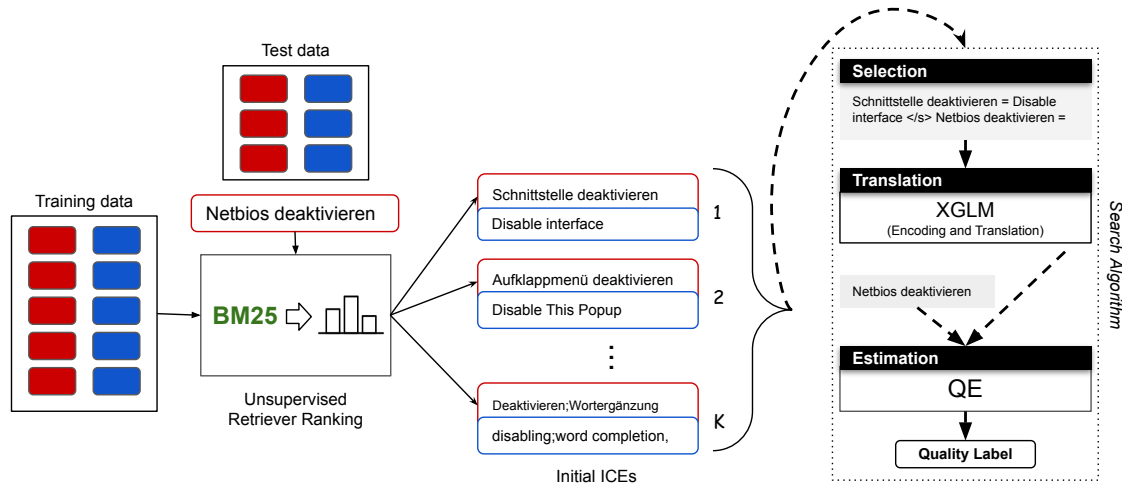
Appendices

Figure 2: **Overview illustration showing an iteration of our proposed methodology.**

Metric	Generic Model	Specific Model
Training Time (hh:mm)	05:55	06:54
CO2 Emissions (kg)	1.41	1.46
Electricity Consumption (kWh)	3.63	3.76

Table 3: **Training Time, CO2 Emissions, and Electricity Consumption for QE Models.**

ICEs:

Die Sockets, die im except Array aufgelistet sind, werden auf Ausnahmen überwacht. = The sockets listed in the except array will be watched for exceptions. $</s>$ Geben Sie den Namen der Variablen ein, deren Wert überwacht werden soll. = Enter the name of the variable whose value is to be monitored. $</s>$ Nur erlaubt bei Sockets für lokale Displays und den globalen Socket. = Permitted only on sockets of local displays and the global socket. $</s>$ Legt fest, ob Scandaten-Information, die in den MPEG2-Videoströmen enthalten sind, aktualisiert werden sollen. = This controls whether to update the scan data information contained in the MPEG-2 video streams. $</s>$ Die Sockets, die im write Array aufgelistet sind, werden daraufhin überwacht, ob ein Schreibvorgang den Socket blockiert. =

Translation:

The sockets listed in the write array will be watched for whether a write operation blocks the socket.

Reference Label:

The sockets listed in the write array will be watched to see if a write will not block.

QE: 67.59, **BLEU score (using reference label):** 52.89

Table 4: **An example of selected ICEs** for a source text, its corresponding translation, reference label, and QE estimation compared to the BLEU score computed based on the reference label.

```
 1: function SEARCH(...)
 2:     temp ← [("", 0.0, "")]
 3:     prompt ← ""
 4:     itr ← 0
 5:     best_qe_score ← 0.0
 6:     patience_counter ← 0
 7:     while itr < iteration and patience_counter < early_stop_patience do
 8:         available_Prompts ← GENERATEAVAILABLEPROMPTS(...)                    ▷ Initial ICEs
 9:         if available_prompts is not empty then
10:             selected_prompt_index ← itr mod k                                ▷ Phase 1: Selection
11:             selected_prompt ← available_prompts[selected_prompt_index]
12:             prompt ← CONSTRUCTFULLPROMPT(...)(see 3.3)
13:             input_ids[0] ← ENCODEPROMPT(...)                                 ▷ Phase 2: Translation
14:             if length(input_ids) > LLM_max_length then
15:                 return temp
16:             end if
17:             output ← GENERATEOUTPUT(...)
18:             final_output ← DECODEOUTPUT(...)
19:             qe_input ← PREPAREQEINPUT(source, final_output)                  ▷ Phase 3: Estimation
20:             qe_score ← ESTIMATEQUALITY(qe_input, model_QE)
21:             temp.append((prompt, current_qe_score, final_output))
22:             if current_bleu_score ≥ 100 then
23:                 return temp
24:             end if
25:             temp ← SORTTEMP(...)
26:             if qe_score ≤ best_qe_score then
27:                 patience_counter ← patience_counter + 1
28:             else
29:                 patience_counter ← 0
30:             end if
31:             best_qe_score ← temp[0][1]
32:         end if
33:         itr ← itr + 1
34:     end while
35:     return temp
36: end function
```

Algorithm 1: **Pseudocode outlining the proposed Search Algorithm.** Each phase of the methodology is annotated alongside the relevant code. Function arguments are omitted for simplicity. The first element of the returning list (*temp*) includes the selected prompt, its associated QE score, and the translated text.