Generative AI in Translation

Alan Zhong
Department of Computer Science
Lakehead University
Thunder Bay, Canada
Group 1
azhong@lakeheadu.ca
1312041

Chomba Waihenya
Department of Computer Science
Lakehead University
Thunder Bay, Canada
Group 1
cwaiheny@lakeheadu.ca
1303398

Dr. Jinan Fiaidhi
Department of Computer Science
Lakehead University
Thunder Bay, Canada
jfiaidhi@lakeheadu.ca

***Abstract*—Generative AI in translation is evolving at an astonishing rate. In this paper, we evaluate the performance of various Generative AI models in translation tasks against high and low resource languages. We also compared them to the state-the-art commercial translators such as MS Translator to further discover the competitiveness of Generative AI models in translation. Our work shows that Generative AI in translation demonstrates good performance in the translation of high resource languages such as in English and Chinese but have degraded results on low resource languages such as Swahili. We further show that hybrid approaches, which combine GPT models with other translation systems, can further enhance the translation quality [1]. Our wish is that this paper will provide an insight into the development of Generative AI especially its key role in translation, that would bring profound benefit to the industry.**

***Index Terms*—GPT, Large Language Models, Translation**

I. Introduction

The advancement in the development of Generative AI is at an unprecedented rate. Nowadays, with an increase in globalization, translation is becoming more essential in numerous industries. Generative AI is making it easier to communicate in different industries by lowering the difficulty of translation through various methods such as increase in translation accuracy, conversational awareness and speed of translation. State-of-the-art models utilize large amounts of pretrained data across different languages and various techniques to derive natural and reliable translations.

Some language pair translations such as Chinese to English pose particular challenges due to fundamental structural differences, contextual reliance for subject identification, and the prevalence of idiomatic phrasing. Generative AI models still demonstrate good performance in the translation of said pairs as long as they are high resource languages. When translating to low resource languages such as Swahili, Generative AI models demonstrate a weaker performance. Commercial translation also experiences the same challenges, and therefore, comparative evaluation is crucial for determining their practical performance.

This paper presents a systematic evaluation of four contemporary translation technologies: DeepSeek, Microsoft Translator, text-davinci-003 model and a hybrid Davinci-based model. Both automated and human evaluation methodologies were applied. BLEU and chrF metrics were used to quantitatively measure lexical and character-level accuracy, while human evaluation assessed fluency, and accuracy. Although COMET was considered due to its strong alignment with human judgment, resource limitations prevented its deployment in our testing environment.

Through this evaluation, we aim to deliver a detailed characterization of the strengths and weaknesses that differentiate commercial translation tools, hybrid and non-hybrid generative AI models. By examining their performance using multiple evaluation metrics and human judgment, this work shows how these systems handle high and low resource language translations. We hope the outcome of our work can provide some guidance for researchers and developers who are seeking reliable and more efficient translation solutions. Furthermore, this comprehensive study may also raise the awareness of the potential for Generative AI's role in translation, which will lead to more support on future innovation in Generative AI multilingual translation at scale.

II. Experimental Setup

A. Dataset

a) WMT Benchmark Data: We employ the most recent WMT (Workshop on Machine Translation), from Chinese(CN) to English(EN) and English(EN) to Chinese(CN) test sets as the primary benchmark dataset. WMT corpora are widely used in MT research because they contain professionally translated parallel text covering multiple genres such as news and conversational writing. The WMT datasets are designed to reduce translationese artifacts and ensure realistic language representations [1] These benchmark datasets enable reliable performance comparison with prior academic work and commercial systems, including the GPT-based evaluations in Hendy et al. (2023) [1].

TABLE I
WMT Dataset Statistics for Chinese–English Translation

Dataset	Direction	# Sentences	Domain Notes
WMT (latest available)	ZH $\rightarrow$ EN	$\sim$1,875	News and mixed domain
WMT (latest available)	EN $\rightarrow$ ZH	$\sim$2,037	News and mixed domain

b) Locally constructed datasets through DeepSeek R1 14b model: To supplement the benchmark evaluation, we constructed a custom dataset using the locally deployed DeepSeek-R1 14B model. We provided a curated set of Chinese and English sentences and collected translations in both directions. Unlike WMT data, these sentences originate from user-created sources and informal contexts, allowing us to evaluate translation robustness across style variations, which further helps with the human evaluations.

TABLE II
WMT DATASET STATISTICS FOR CHINESE–ENGLISH TRANSLATION

Dataset	Direction	# Sentences	Domain Notes
DeepSeek-R1	ZH → EN	∼50	Informal
DeepSeek-R1	EN → ZH	∼50	Informal

c) Justification for custom dataset from DeepSeek: The WMT benchmark serves as a controlled test environment where model outputs can be directly compared to prior studies by the original researchers. It is professionally translated and clearly defined reference standards [1]. In contrast, the dataset generated by DeepSeek introduces more informal, conventional, and conversational sentence structures. By evaluating performance across these complementary settings, our study captures a more complete view of how current commercial translator (Microsoft Translator) and generative AI models behave in practical usage scenarios. This design aligns with recommendations in recent literature evaluating GPT-based translation capabilities, which emphasize the importance of stress testing models beyond traditional benchmark datasets. The implementations will be shown later in the experiments section.

d) Ethical and Data Usage Information: All benchmark data used in this study is publicly available and widely licensed for research purposes. The DeepSeek generated dataset contains only translations of text we created specifically for this project and does not include any personally identifiable or sensitive information. No copyrighted third-party content was used to construct evaluation data.

B. Environment

All experiments were conducted on a Windows 11 machine with the Anaconda Distribution for Python environment management. The software environment consisted of Python 3.10, Git, and Poetry for dependency management. The project utilized the following packages and libraries: SentencePiece (v0.1.96), Pandas (v1.4.1), Transformers (v4.17), PyTorch Lightning (v1.6.4), JSONArgParse (v3.13.1), NumPy (v1.20.0), TorchMetrics (v0.8.2), SacreBLEU (v2.0.0), SciPy (v1.5.4), Entmax (v1.1), Sphinx-Markdown-Tables (v0.0.15), Coverage (v5.5), and Scikit-Learn (v0.24). A CPU-compatible version of PyTorch was used, installed from the official wheel: torch = {url = "https://download.pytorch.org/whl/cpu/torch-2.0.0%2Bcpu-cp310-cp310-win_amd64.whl"} This choice was required due to GPU incompatibility with the version of PyTorch utilized in the original paper.

The software environment was prepared using the following steps:

- Install all required dependencies and packages listed above.
- Clone the repository for the paper:
  git clone https://github.com/microsoft/gpt-MT.git
- Navigate to the tools directory:
  cd tools
- Create a dedicated Anaconda environment:
  conda create -n gpt-mt-eval python=3.10
- Activate the environment:
  conda activate gpt-mt-eval
- Upgrade pip:
  pip install –upgrade pip
- Install the base requirements:
  pip install -r requirements.txt
- Clone the COMET evaluation repository:
  git clone https://github.com/Unbabel/COMET.git
- Enter the COMET directory:
  cd COMET
- Check out the required branch:
  git checkout fc2f2b3
- Modify the pyproject.toml file under tool.poetry.dependencies by replacing the Torch dependency with the CPU-based wheel:
  torch = {url = https://download.pytorch.org/whl/cpu/torch-2.0.0%2Bcpu-cp310-cp310-win_amd64.whl}

  This ensures use of the CPU version of PyTorch. Although the original paper used GPU-based PyTorch, GPU support could not be configured due to hardware incompatibility, and attempting to upgrade the GPU version resulted in source-level errors.
- Resolve dependency conflicts created by the change:
  poetry lock
- Install all required packages and dependencies:
  poetry install

C. Local Language Model Management

Ollama was chosen as the model deployment and management framework in this project. Ollama provides a lightweight runtime environment for running large language models locally, it also enables direct interaction with the models through a command-line interface without requiring cloud-based compute resources. Ollama is

installed from the official distribution package available at: `https://ollama.com/download` After installation, users can obtain a variety of open-source LLMs from the official model registry: `https://ollama.com/search` Models can be downloaded and executed locally using the command:

```
ollama run <model-name>
```

For this project, we specifically deploy the DeepSeek-R1-14b model to generate translation outputs in both Chinese→English and English→Chinese directions. Local deployment allows us to:

- Ensure improved reproducibility and model version consistency
- Avoid network latency and rate-limiting constraints from hosted APIs
- Evaluate model behavior independent of external service optimizations

This setup also facilitates the construction of our custom evaluation dataset described in Section 2.1.

D. Evaluation Methods

To assess translation quality, we chose a combination of automatic and human-centered evaluation approaches to capture different aspects of performance.

a) BLEU: Bilingual Evaluation Understudy, is a reference-based metric that measures the similarity between system output and professional ground-truth translations using modified n-gram precision with a brevity penalty. If the translated sentence length is shorter than the original sentence, it receives a penalty.

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{1}$$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - \frac{r}{c}) & \text{if } c \leq r \end{cases} \tag{2}$$

where:

- p_n is the modified precision for n-grams.
- w_n is the weight assigned to each n-gram order (typically $w_n = \frac{1}{N}$ for uniform weighting).
- BP is the brevity penalty applied to avoid rewarding excessively short translations.
- c is the total length of the candidate translation.
- r is the total length of the reference translation.

b) chrF: CHaRacter-level F-score evaluates translation quality based on character-n-gram F-score, which improves sensitivity to morphology and variations in the word's form, particularly for languages without explicit word spacing.

$$\text{chrF}_\beta = \frac{(1 + \beta^2) \cdot (\text{Precision} \cdot \text{Recall})}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{3}$$

where:

- Precision is the character-level n-gram precision score.
- Recall is the character-level n-gram recall score.
- β controls the relative importance of recall over precision (commonly $\beta = 2$ to emphasize recall).

c) COMET: We explored the use of COMET, a state-of-the-art neural MT evaluation metric designed to align closely with human judgments by leveraging multilingual encoders and reference-based scoring models[1]

COMET has gained prominence for reliable quality estimation due to its ability to evaluate meaning preservation rather than n-gram overlap.

However, due to the lack of available GPU resources within our local environment, COMET evaluation could not be performed. Limited execution attempts resulted in unstable processing and fetal crashes on our workstations. Consequently, we had to shift our focus to the BLEU, chrF, and human-based evaluation methods.

We highlight COMET's omission as a limitation and an opportunity for future work.

III. Experiments

A. DeepSeek Translation Procedures

To perform local translations, we deployed the DeepSeek-R1-14b model through the Ollama runtime environment. The model is executed using:

```
ollama run deepseek-r1:14b
```

After initialization, the interface becomes interactive once the command prompt displays the indicator:

```
>>>
```

The designation "14b" refers to the model size, representing approximately 14 billion trainable parameters. Larger parameter count generally correlates with improved semantic understanding and translation quality, but also requires increased hardware resources for efficient execution, in some cases the increments are significant.[2] We perform translation by issuing natural-language prompts in a zero-shot setting. The standard prompt template is:

```
Translate "<sentence>" into <target_language>.
```

The model automatically identifies the source language and generates the translation in the specified target language without requiring additional configuration or parallel exemplar prompts. Both Chinese→English and English→Chinese, Swahili→English and English→Swahili directions were tested using this uniform prompting strategy for consistency. This setup enables us to directly compare the local DeepSeek model's translation capabilities against commercial cloud-based systems and other generative AI models under equivalent input conditions.

TABLE III
COMPARISON OF CHRF AND BLEU SCORES ACROSS TRANSLATION SYSTEMS

Model	Language Pair	chrF	BLEU	Language Pair	chrF	BLEU
3*MS Translator	cs→en	74.0	54.9	en→cs	65.6	42.1
	fr→de	67.5	45.3	de→fr	65.0	42.0
	zh→en	57.7	27.9	en→zh	43.1	48.1
3*DeepSeek-R1 14b (Local)	sw→en	14.5	0.0	en→sw	45.7	26.3
	zh→en	56.5	60.7	en→zh	42.9	49.8
	(Other language pairs not available)					
3*text-davinci-003	cs→en	67.5	44.5	en→cs	57.9	31.3
	fr→de	65.7	42.5	de→fr	58.9	35.6
	zh→en	56.0	25.0	en→zh	34.6	38.3
2*Hybrid text-davinci-003	cs→en	72.3	52.3	en→cs	64.0	39.6
	zh→en	56.5	25.8	en→zh	42.2	46.5

B. Evaluation Script

The evaluation script provides a command line interface for computing machine translation quality scores across multiple language directions. To view all available configuration options, the following command may be executed:

```
python evaluate.py -h
```

For the experiments in this work, the parameters listed below are required:

```
--testset TESTSET:
 Specifies the path to the test set directory.
      The directory must contain, for each
     language direction, reference and source
     files following the structure:
 {src_lang}{tgt_lang}/test.{src_lang}-{
     tgt_lang}.{tgt_lang} (target reference)
     and
 {src_lang}{tgt_lang}/test.{src_lang}-{
     tgt_lang}.{src_lang} (source input).
 These pairs are used by the evaluation script
      to align system outputs with their
     corresponding gold-standard references.

--hypotheses HYPOTHESES:
 Indicates the path to the model-generated
     translation outputs. For each evaluated
     direction, this directory must contain:
 {src_lang}{tgt_lang}/test.{src_lang}-{
     tgt_lang}.{tgt_lang}
 This file is compared directly against the
     reference to compute metric scores.

--directions DIRECTIONS:
 Lists the language pairs to be evaluated (e.g
     ., "en-de de-en").
 Each direction corresponds to a pair of
     source and target languages for which both
      references and system outputs must be
     available.

--metrics METRICS:
 Specifies the evaluation metrics to be
     computed. In this work, we use BLEU and
     chrF, two standard metrics widely used in
     machine translation evaluation.

--save-name SAVE_NAME:
 Provides a name for the output directory in
     which the evaluation results will be
     stored.
```

An example of how the evaluation script is run is shown below:

```
python evaluate.py \
   --testset ../evaluation/testset/wmt-testset
        \
   --directions de-en en-de cs-en en-cs ja-en
       en-ja zh-en en-zh \
             ru-en en-ru uk-en en-uk is-en en-
                 is ha-en en-ha fr-de de-fr \
   --metrics chrf bleu \
   --hypotheses ../evaluation/system-outputs/
       text-davinci-003/QR/5-shot
```

C. Experiment 1 (MS Translator)

Microsoft Translator demonstrates strong baseline performance across all tested language pairs. For CS↔EN and FR↔DE, both BLEU and chrF remain consistently high, indicating reliable translation fluency and lexical alignment with references. Performance remains solid for ZH↔EN, although BLEU scores for ZH→EN drop compared to European language pairs. This degradation is expected due to the larger structural and morphological differences between Chinese and English. EN→ZH BLEU shows competitive results, suggesting that the system may be optimized more effectively when generating Chinese output from English. Overall, Microsoft Translator provides stable, high-quality translation results for high resource languages. Results shown in Table III.

D. Experiment 2 (DeepSeek)

The locally deployed DeepSeek model shows mixed performance. For ZH↔EN, the model performs competitively with BLEU scores reaching 60.7 for ZH→EN, the highest value observed among all tested models for this direction. chrF values also remain comparable to commercial systems. However, performance on Swahili - English (used for stress - testing generalization) is significantly lower, with BLEU scores falling near zero for sw→en. This is expected as DeepSeek-R1-14b is just a small model compared with its other variations (32b, 64b, Zero etc). The result also proves that Generative AI struggles with low resource languages compared with high

resource languages. Local deployment still proves effective for generating usable CN - EN translations without reliance on cloud services. Results shown in Table III.

E. Experiment 3 (text-003-davinci)

The text-davinci-003 model provides average translation quality across the benchmark language pairs. Performance is lower than Microsoft Translator, reflecting that it was not explicitly optimized for machine translation tasks. Still, the results confirm that general purpose generative models can perform translation at a usable level without specialized training or parallel data. An interesting observation is that quality varies significantly with direction, with EN→ZH often outperforming ZH→EN which is a trend mirrored in commercial systems as well. Results shown in Table III.

F. Experiment 4 (Hybrid-text-davinci-003)

Results show consistent improvements over standalone text-davinci-003 result, particularly for CS↔EN, where BLEU increases from 44.5 → 52.3. This demonstrates the benefit of system level interventions that compensate for variance in generative model performance. However, gains are smaller for ZH↔EN, and in some cases chrF remains unchanged. The approach successfully reduces error cases without requiring GPU acceleration or model fine tuning. Results shown in Table III.

IV. Conclusion

Our results demonstrate that hybrid systems can yield improved translation performance. However, translation accuracy decreases substantially for low resource languages, highlighting data scarcity as a key factor limiting generalization in current Generative AI in translation models. In contrast, high resource language pairs continue to benefit from abundant multilingual training data, producing more fluent and accurate translations across all systems tested.

V. Limitations and Future Work

Due to hardware limitations, we were unable to get COMET scores for translations, and deploy more complicated LLM models on our machines. We have also noticed the lack of language pair tests such as FR - EN in the original dataset provided by the original researchers. In the future, if the budget allows, we would like to continue the research to make it more complete.

References

[1] Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla, "How Good Are GPT Models at Machine Translation? A Comprehensive Evaluation," arXiv preprint arXiv:2302.09210, 2023.

[2] X. Zane, "Difference between DeepSeek 14B and 32B explained," BytePlus, Aug. 22, 2025. [Online]. Available: https://www.byteplus.com/en/topic/385133?title=difference-between-deepseek-14b-and-32b-explained. [Accessed: Nov. 29, 2025].