Combining the Best of Both Worlds: A Method for Hybrid NMT and LLM Translation

Anonymous ACL submission

Abstract

Large language model (LLM) shows promising performances in a variety of downstream tasks, such as machine translation (MT). However, using LLMs for translation suffers from high computational costs and significant latency. Based on our evaluation, in most cases, translations using LLMs are comparable to that generated by neural machine translation (NMT) systems. Only in particular scenarios, LLM and NMT models show respective advantages. As a result, integrating NMT and LLM for translation and using LLM only when necessary seems to be a sound solution. A scheduling policy that optimizes translation result while ensuring fast speed and as less LLM usage as possible is thereby required. We compare several scheduling policies and propose a novel and straightforward decider that leverages source sentence features. We conduct extensive experiments on multilingual test sets and the result shows that we can achieve optimal translation performance with less LLM usage, demonstrating effectiveness of our decider.

Figure 1: A comparison of translations done by an NMT model and LLM. They translate simple content equally well but their performances vary when translating complex sentences.

1 Introduction

Neural models (Sutskever et al., 2014; Bahdanau et al., 2016; Vaswani et al., 2017) greatly boost machine translation (MT) performance while various inference speed-up strategies (Wang et al., 2019, 2021b,a) ensure fast translation. As model scales, large language model (LLM) (Ouyang et al., 2022; Touvron et al., 2023) now is able to deliver fairly good translation results(Son and Kim, 2023; Zhang et al., 2023; Moslem et al., 2023), which are even comparable to translations done by commercial translation systems. Hendy et al. (2023) find that LLM performs particularly well when translating content in particular domains. As shown in Figure 1, we find that neural machine translation (NMT) model and LLM have own merits and drawbacks.

Methods to harness complementary strengths of NMT and LLM models, with the aim of achieving better translation results through their integration, are worthy of research. To integrate large and small models, Zeng et al. (2024) propose Cooperative Decoding while Farinhas et al. (2023) put forward an approach based on Minimum Bayes Risk (MBR). While these methods are effective in improving translation quality, their reliance on LLMs for every piece of translation incurs substantial computational expenses.

As shown in Table 1, we annotate source sentences into two categories: simple and hard (to translate). The majority of sentences are marked as simple, and only a small portion of sentences is believed to be challenging for translation systems. LLM performs better when translating complex sentences. (Hendy et al., 2023) perform translation using an NMT model and evaluate the translation quality using a quality estimation (QE) model (Rei et al., 2020; Fomicheva et al., 2020; Rei et al., 2022c). If the QE model give a low score to a translation result, they then use an LLM to translate the sentence again. Their approach leverages LLM advantages while reduce unnecessary LLM usage.

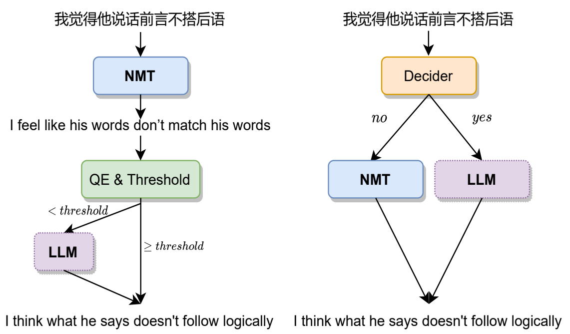

Figure 2: Two approaches to integrate NMT model and LLM. The left approach is QET proposed by (Hendy et al., 2023) and the right part is our proposed PPLT and JDM, which quickly determines when to use LLM based on source sentence.

However, their approach heavily relies on the performance of the QE model, and ignores scenarios when LLM deliver even worse results. As a result, their approach may not guarantee best translation outcome.

Without using any QE model, our approach decides when to use LLM only based on source sentence features. This idea is challenging and we try a multiple of indicators. In the end we find that using only two indicators–sentence complexity and translation domain (whether LLM is good at or not), we can make a sound decision. In this way, we can directly decide whether to use NMT model or LLM as long as the source sentence is input, and use LLM as less as possible. We test our approach on multilingual test sets (Zh2En, En2Zh, De2En, and Ja2En) and obtain best in results for MT .

2 Method

As shown in Figure 2, compared to the QE Threshold (QET) method proposed by Hendy et al. (2023), our approach is more straightforward. The former requires using the wmt22-cometkiwi-da[1] (Rei et al., 2022b) model to assess the quality of NMT translation results to decide whether to continue calling the LLM for translation. In contrast, our method directly decides whether to call the LLM or NMT based on the input source text, which is clearly a challenging task. Moreover, our method needs to meet two requirements: 1) minimize the use of LLMs as much as possible, and 2) if LLM results are used, they should outperform the NMT results. It is evident that using QET makes it difficult to fully satisfy the second condition, which impacts the final fusion effect.

[1]https://huggingface.co/Unbabel/wmt22-cometkiwi-da

	$DA_{Simple}(95\%)$	$DA_{Hard}(5\%)$
NMT	80.21	73.22
LLM	81.62	77.02
Diff	**1.41**	**3.80**

Table 1: Comparison of NMT and LLM performances on simple and complex sentences in the WMT22 Zh2En news test set. "Diff" refers to the difference in DA scores between LLM and NMT. 95% of the sentences are considered easy to translate. We detail our classification criteria in Appendix B. We conduct experiment using an NMT model trained from scratch and Llama-3.1-8B-Instruct (Touvron et al., 2023). wmt22-comet-da (Rei et al., 2022a) is used for reporting DA score.

As shown in Table 1, LLM delivers better translation when source sentence is hard to translate. For relatively simple sentences, LLM does not have a particular advantage, so we try not to use LLM in this scenario. In addition, we need to verify whether it is possible to determine using which model to translate only based on source sentence. As a result, we design two approaches to meet the two requirements:

PPL Threshold (PPLT): We use monolingual data (previously used for training NMT models) to train a small language model (LM). We directly use LLM for translation when the source sentence perplexity (PPL) is greater than a threshold we set. We employ the simple method to test whether LLM can translate complex sentences well.

Joint Decision-making (JDM): Given a source sentence (src), we use NMT model and LLM to obtain two translation results (tgt_{NMT} and tgt_{LLM}). By comparing the two translations against reference (tgt), we obtain quality measurements of the two results (Q_{NMT} and Q_{LLM}). We hope to use LLM for translation only when (1) the translation delivered by NMT model is bad and (2) the translation done by LLM is better. So we use LLM for translation when:

$$Q_{NMT} < T_1 \; and \; Q_{LLM} - Q_{NMT} > T_2 \quad (1)$$

Due to the inability to obtain references in practical applications, we cannot directly use the above conditions to control the LLM's invocation. Therefore, based on these conditions, we select positive and negative samples from bilingual data and train a binary classification model to serve as a decider, determining when to use the LLM for translation during the inference process.

	News			Flores			Literary			Tech			Avg		
	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p
NMT	78.99	66.12	0.00%	87.08	76.10	0.00%	59.71	46.29	0.00%	83.38	71.97	0.00%	77.29	65.12	0.00%
LLM	80.13	67.26	100.00%	86.68	75.55	100.00%	66.69	53.86	100.00%	77.68	61.27	100.00%	77.80	64.49	100.00%
QET	79.13	66.25	20.32%	87.08	76.11	0.30%	63.88	50.62	62.00%	80.21	66.14	39.60%	77.58	64.78	30.55%
PPLT	79.24	66.14	38.19%	87.06	76.08	5.43%	63.49	50.73	51.40%	82.02	69.37	34.20%	77.95	65.58	32.31%
JDM	**79.69**	**66.91**	29.39%	**87.12**	**76.12**	1.28%	**65.70**	**52.70**	80.40%	**82.71**	**70.88**	7.00%	**78.81**	**66.65**	29.52%
oracle	*82.25*	*69.86*	56.91%	*88.17*	*77.59*	48.32%	*68.41*	*54.54*	72.80%	*84.80*	*73.14*	28.20%	*80.91*	*68.78*	51.56%

Table 2: Integration performances of QET, PPLT, and JDM methods on four Chinese→English test sets.

	News			Flores			Literary			Tech			Avg		
	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p
NMT	86.17	71.89	0.00%	87.88	73.29	0.00%	71.68	51.83	0.00%	86.30	73.57	0.00%	83.01	67.65	0.00%
LLM	85.17	68.20	100.00%	86.63	68.91	100.00%	76.30	56.12	100.00%	78.40	59.74	100.00%	81.63	63.24	100.00%
QET	86.08	71.52	8.15%	**87.83**	**73.14**	2.77%	72.73	52.76	24.31%	80.57	64.02	53.40%	81.80	65.36	22.16%
PPLT	85.80	70.76	28.47%	87.73	72.61	21.15%	72.85	53.25	25.37%	85.02	71.05	27.00%	82.85	66.92	25.50%
JDM	**86.18**	**71.59**	20.77%	87.76	72.92	9.19%	**75.15**	**54.97**	53.91%	**85.39**	**71.97**	9.60%	**83.62**	**67.86**	23.37%
oracle	*88.00*	*73.16*	38.05%	*89.03*	*73.73*	36.26%	*78.16*	*58.42*	63.85%	*87.11*	*73.74*	20.80%	*85.58*	*69.76*	39.74%

Table 3: Integration performances of QET, PPLT, and JDM methods on four English→Chinese test sets.

3 Experiments

3.1 NMT & LLM

We directly use Llama-3.1-8B-Instruct as our LLM model, and its translation prompt is provided in Appendix D. Due to its strong translation performance, we do not perform any further supervised fine-tuning on it. We focus on training an NMT model from scratch for each language pair that can achieve comparable translation performance. Our NMT model adopts the Deep Transformer-Big architecture commonly used by Wei et al. (2022), and the training data comes from internal technology (Tech) bilingual data and various open-source bilingual datasets archived in OPUS, such as CCMatrix, Paracrawl, NLLB, UNPC and OpenSubtitles. For each translation language pair, we randomly sample 100 million bilingual data, which are then deduplicated with the test set before being used for training. The training setup of NMT models is provided in Appendix E.

3.2 Threshold

Among the methods mentioned above, all require setting thresholds in advance to limit the proportion of LLM calls. For each language pair, we use statistical methods to determine the threshold for each method by controlling the proportion of LLM calls to approximately 25%, and then compare the performance differences between different methods. The specific threshold values for each method are provided in Appendix F.

For the QET method, it requires setting a QE score threshold to control the invocation of the LLM. The LLM is only called when the QE score of the NMT translation is below this threshold. We select one million bilingual data, obtain the NMT translation corresponding to the source text, and calculate their QE scores using wmt22-cometkiwi-da (Rei et al., 2022b). The QE scores are then sorted in ascending order, and the 250,000th QE score in the sorted list is used as the threshold.

For the PPLT method, it is necessary to set a PPL score threshold, and the LLM is invoked only when the source text's PPL score exceeds this threshold. We calculate the PPL scores of 1 million source-language monolingual data using a self-trained LM model, then sort the PPL scores in descending order and use the 250,000th score in the sorted list as the threshold. The LM is trained on 30 million monolingual data, with training settings provided in Appendix E. Additionally, all monolingual data is randomly sampled from bilingual data.

For the JDM method, when to call the LLM depends on the decision maker, which is a binary classification model fine-tuned based on xlm-roberta-base (Conneau, 2019). The training setup is described in Appendix E. When selecting positive and negative samples for training according to Equation 1, two thresholds need to be set. Specifically, we use one million bilingual data, obtain the NMT translation and LLM translation corresponding to the source text, and calculate the Q_{NMT} and Q_{LLM} scores using wmt22-comet-da [2] (Rei et al., 2022a). The Q_{NMT} scores are then sorted from lowest to highest, and the 100,000th score in the sorted list is chosen as the T_1 threshold. Then, the data with Q_{NMT} scores lower than T_1 are sorted by the difference between Q_{LLM} and Q_{NMT} scores, from highest to lowest, and the 10,000th score is

[2] https://huggingface.co/Unbabel/wmt22-comet-da

selected as the T_2 threshold. Ultimately, we can obtain 10,000 positive samples, and then randomly select 30,000 negative samples from the remaining data. With this training data ratio, the decider can maintain about 25% LLM calls.

3.3 Test Set

We use WMT22 News and Flores (Costa-jussà et al., 2022) test sets for all language pairs we selected. To better test our method on Zh2En and En2Zh, we construct a Literary test set and a Tech test set. Each test set contains 500 sentences. We find that LLM's performance is much better on the Literary test sets while NMT models outperform on the Tech test sets. These test sets can better evaluate the effectiveness of different fusion strategies. We will open-source these two self-constructed test sets to promote the development of NMT and LLM fusion technologies.

4 Results

We validate the aforementioned methods on four Zh↔En test sets, as shown in Table 2 and Table 3. wmt22-comet-da (Rei et al., 2022a) is used for reporting DA score (%). BLEURT20[3] (Sellam et al., 2020) is used for reporting BLEURT score (%). LLM_p refers to the percentage of LLM usage. We also report the performance of oracle system that selects the best translation results based on wmt22-comet-da, representing the upper bound that the fusion method can achieve. In terms of the two base models, NMT and LLM, their performance gap is small on the open-source news and Flores test sets, but there are significant differences on the Literary and Tech test sets.

QET reduces LLM usage to 30.55% on Zh2En test sets and 22.16% on En2Zh test sets. Regarding translation quality, its performance remains almost the same as the optimal single system result on the Zh2En test sets. But We witness one point down of DA score and BLEURT on the En2Zh test sets when comparing with the optimal single system result (NMT), because QET performs not so well on the Tech test set. Our NMT model significantly outperforms LLM on the Tech test set but QET integrate some worse LLM translations to the final results. QET uses LLM when NMT translation quality is poor, but it does not evaluate whether LLM's translation is better or even worse.

[3] https://github.com/google-research/bleurt

Interestingly, our PPLT method achieved better results than the QET method, while the number of LLM calls was only slightly higher. The result demonstrates that using merely source text features, i.e. text complexity, can get a desirable integration of NMT model and LLM. In addition, as LM is trained on NMT-similar data, PPLT significantly outperforms QET on the Tech test sets.

Our JDM method achieves best performance on average. It seems that JDM method is able to dynamically control LLM usage as it varies greatly on different test sets. For instance, LLM outperforms NMT model on News and Literary test sets, LLM usage is thus high (even over 50% on the Literary test sets). On contrary, NMT model outperforms LLM on the Tech test sets, so LLM usage is greatly decreased (less than 10%). Regarding domains where LLM and NMT model's performances are equal, JDM uses LLM as less as possible. On the Tech and Flores test sets where NMT model outperforms, we witness more LLM usage on the Tech test sets than on Flores, because the Tech test sets are more difficult to translate (6-8 points gap regarding DA score), and the decider tends to pass complex sentences to the LLM.

Similar results prevail on De2En and Ja2En test sets (see Appendix G). When LLM and NMT each have their respective advantages, JDM can also achieve better integration results.

5 Conclusion

This paper proposes a fast approach to integrate LLM and NMT model in order to improve translation quality while ensuring fast speed and low cost. Compared with previous methods that use QE models, our decider determines when to use LLM based on source sentence features. We train an end-to-end decider to get the desired performance, that is, use LLM as less as possible and use LLM for translation only when it outperforms NMT model. We test our approach on multiple test sets, including Zh2En, En2Zh, De2En, and Ja2En. The result shows that this straightforward approach achieves almost the best performance. In addition, our experiments show that LLM has a particular advantage over NMT when translating internet memes and informal expressions (new concepts consists of high-frequency words).

Limitations

We propose a simple and fast method to integrate NMT and LLM, and experiments verify the effectiveness of our method. However, we find that the final integration performance depends on the complementarity between NMT and LLM. That is to say, only when the NMT model and LLM has respective advantages can the integration lead to better translations. If the NMT performs equally as LLM, we see no improvement after integration.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate. *ICLR*.

A Conneau. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

António Farinhas, José de Souza, and Andre Martins. 2023. An empirical study of translation hypothesis ensembling with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11956–11970, Singapore. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *TACL*.

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. *arXiv preprint arXiv:2302.09210*.

Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. *EAMT*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *NIPS*.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022a. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

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.

Ricardo Rei, Marcos Treviso, Nuno M Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José GC De Souza, Taisiya Glushkova, Duarte M Alves, Alon Lavie, et al. 2022b. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. *arXiv preprint arXiv:2209.06243*.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte M. Alves, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022c. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. *Preprint*, arXiv:2209.06243.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Jungha Son and Boyoung Kim. 2023. Translation performance from the user's perspective of large language models and neural machine translation systems. *Information*, 14(10).

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *NIPS*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NIPS*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. *ACL*.

Xiaohui Wang, Ying Xiong, Xian Qian, Yang Wei, Lei Li, and Mingxuan Wang. 2021a. Lightseq2: Accelerated training for transformer-based models on gpus. *arXiv preprint arXiv:2110.05722*.

Xiaohui Wang, Ying Xiong, Yang Wei, Mingxuan Wang, and Lei Li. 2021b. LightSeq: A high performance inference library for transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers (NAACL-HLT)*, pages 113–120. Association for Computational Linguistics.

Daimeng Wei, Zhiqiang Rao, Zhanglin Wu, Shaojun Li, Yuanchang Luo, Yuhao Xie, Xiaoyu Chen, Hengchao Shang, Zongyao Li, Zhengzhe Yu, et al. 2022. Hw-tsc's submissions to the wmt 2022 general machine translation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 403–410.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jiali Zeng, Fandong Meng, Yongjing Yin, and Jie Zhou. 2024. Improving machine translation with large language models: A preliminary study with cooperative decoding. *Findings in ACL*.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. *Preprint*, arXiv:2301.07069.

A Advantages of LLM in Translation

Our experiments demonstrate that when source sentences are simple, NMT model and LLM have similar performances. When source sentences are hard to translate, LLM outperforms NMT model, and the gap becomes larger on extremely complex sentences. Our proposed decider uses LLM for translation only when LLM performs better, thus ensuring minimum LLM usage. We manually collect complex sentences based on three criteria described in Appendix B. We send these complex sentences to our decider to see how the decider allocates these sentences, so we can see whether LLM is indeed better on complex content.

We select 50 test cases under each criterion and send them to the decider. As shown in Table 4,

Type	LLM_p	Samples
category 1	68%	50
category 2	42%	50
category 3	5%	50

Table 4: The proportion of LLM being used under three different types of hard-to-translate categories.

LLM's performance varies under the three categories. For category 1 (informal expression or Internet memes), 68% test cases can be well translated by LLM. For category 2 (specialized terms or expressions), only 42% can be well translated. However, LLM also fails on category 3 (context is required for translation), as only 5% cases can be well handled.

B Difficult Text Types

For category 1, NMT models can hardly translate those informal expressions and memes correctly. However, a majority of those informal expressions and memes are new combinations of high-frequency words, which may not be rare in LLM training data. Moreover, as model scales, LLM can better grasp the true meaning of source sentences. So LLM can deliver relatively good translations. However, for category 2, specialized terms are low-frequency words in both NMT and LLM training data, so LLM also struggles to translate those terms correctly. For category 3, without context, LLM also fails if we use it as a sentence-level translation system.

We define hard-to-translate Chinese sentences from three dimensions: (1) Sentences containing informal expressions or Internet memes, of which the literal meaning is wrong or misleading. For example

Example: 浙大学术年会上学生唱主角研究成果让人脑洞大开

Reference: Students played the leading role at the Annual Academic Conference of Zhejiang University with creative research achievements

Explanation: "脑洞大开" literally means "a big hole in the brain" but its actual meaning is "creative; inspiring".

(2) Sentences containing specialized terms or expressions, requiring domain knowledge to understand the true meaning.

Example: 桃胶的作用：味苦，性平，归大肠，膀胱经。

Reference: Functions of peach glue: bitter in taste, mild in nature, belong to the large intestine and bladder channels.

Explanation: "性平" is a term in traditional Chinese medicine. "性" means the intrinsic properties of herb, including cold, hot, warm, cool, or mild (平).

(3) Sentences that cannot be understood without context:

Example: 女孩街头“箭靶”募捐被告诫

Reference: The girl who raised donations by acting as a target on the street was warned.

Explanation: "箭靶" literally means "archery target. If it is translated literally, the sentence would be "The girl made an archery target donation on the street and was warned.", But according to context, it actually means "the girl acted as a target".

C Literary Test Set

The composition of the literary test set:

(1) Sentences require transcreation: needs to recreate content in a new language and maintain its original meaning, such as slogan, and advertisements.

(2) Sentences contain memes and buzzwords: needs to understand local popular culture before translation.

(3) Sentences contain idioms: needs to understand local literature and social culture before translation.

Test cases are crawled from English learning websites. References are double-checked by in-house translators.

D LLM Translation Prompt

Figure 3 illustrates the translation prompt used for LLM. {source_language} and {target_language} denote the full names of the languages involved, for example, "Translate this from Chinese to English." {source_sentence} represents the content that actually needs to be translated.

E Training Setup

We use the open-source fairseq (Ott et al., 2019) to train NMT models. The key parameters are as follows: each model is trained on 8 GPUs, with a batch size of 6144 and a parameter update frequency of 2. The learning rate is set to 5e-4. The warm-up steps are set to 4000, and the model is checked every 1000 steps. Additionally, we apply a dropout rate of 0.1 and use R-Drop with default hyperparameters. Training is stopped when the evaluation metrics on the development set do not improve for 10 consecutive checkpoints. The last 10 saved models are averaged and then used for translation.

We also use the open-source fairseq (Ott et al., 2019) to train LM models. The model architecture follows transformer_lm_base, and the other main training parameters are consistent with those used for the NMT model. Training is stopped when the loss on the development set does not decrease after 10 consecutive checkpoints, and the last saved model is used to compute PPL.

We use the open-source transformers (Wolf et al., 2020) to train deciders for each language pair. The model architecture consists of a linear layer connected to a pretrained LM. The key training parameters are as follows: the learning rate for the pretrained LM is set to 1e-5, the learning rate for the linear layer is set to 1e-3, the batch size is 32, the gradient accumulation steps are 8, and the dropout rate is set to 0.3. We only train for one epoch, and the last saved model is used for decision-making during the inference phase.

F Threshold Value

The threshold values of the various methods we used are shown in Table 5. QET and PPLT threshold values directly control the invocation frequency of LLMs, while JDM T_1 and T_2 are used to select training samples for the decider, indirectly affecting the invocation frequency of LLMs.

	QET	PPLT	JDM T_1	JDM T_2
Zh2En	70	5.6	73	3.5
En2Zh	72	5.5	76	3.5
De2En	67	5.7	79	2.5
Ja2En	73	5.8	64	3.5

Table 5: The threshold values of the various methods.

G De2En and Ja2En Results

We also validate the fusion effects of the QET and JDM methods on De2En and Ja2En. To better highlight the characteristics of these two methods, we include the Subtitle and Travel test sets. These two test sets are chosen because there is a noticeable gap in the translation results between NMT and LLM models, which helps to better evaluate the effects of system fusion. The Subtitle and Travel test sets are collected from open-source data such as OpenSubtitles, JESC, TED, QED, and czechtourism, and then constructed by language experts, with each test set containing 500 sentences.

<|begin_of_text|><|start_header_id|>system<|end_header_id|>\n\n
Translate this from {source_lang} to {target_lang}:<|eot_id|>
<|start_header_id|>user<|end_header_id|>\n\n
{source_lang}:{source_sentence}<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>\n\n{target_lang}:

Figure 3: LLM Translation Prompt

	News			Flores			Subtitle			Travel			Avg		
	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p
NMT	79.15	64.29	0.00%	87.76	75.51	0.00%	84.92	74.31	0.00%	79.00	65.16	0.00%	82.71	69.82	0.00%
LLM	80.76	66.52	100.00%	87.43	74.59	100.00%	82.41	70.25	100.00%	83.80	69.37	100.00%	83.60	70.18	100.00%
QET	79.17	64.39	15.49%	87.77	75.52	0.30%	83.21	72.13	46%	80.74	66.47	26%	82.72	69.63	21.95%
JDM	**80.28**	**65.80**	39.39%	**87.81**	**75.53**	2.27%	**84.02**	**73.64**	33%	**83.13**	**68.62**	15%	**83.81**	**70.90**	22.42%

Table 6: Integration performances of QET and JDM methods on four Japanese→English test sets.

	News			Flores			Subtitle			Travel			Avg		
	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p	DA	BLEURT	LLM_p
NMT	83.53	71.84	0.00%	89.20	79.83	0.00%	87.12	77.87	0.00%	87.72	77.09	0.00%	86.89	76.66	0.00%
LLM	84.72	73.27	100.00%	89.19	79.59	100.00%	86.28	76.05	100.00%	90.87	82.10	100.00%	87.77	77.75	100.00%
QET	83.64	71.97	8.67%	89.20	79.83	1.58%	86.55	76.82	43.00%	88.61	78.63	35.00%	87.00	76.81	22.06%
JDM	**84.31**	**72.74**	36.39%	**89.29**	**79.93**	10.87%	**86.90**	**77.24**	29.00%	**90.24**	**80.93**	17.00%	**87.69**	**77.71**	23.32%

Table 7: Integration performances of QET and JDM methods on four German→English test sets.