Anticipating Future with Large Language Model for Simultaneous Machine Translation

Siqi Ouyang[1], **Oleksii Hrinchuk**[2], **Zhehuai Chen**[2], **Vitaly Lavrukhin**[2], **Jagadeesh Balam**[2], **Lei Li**[1], **Boris Ginsburg**[2],
[1]Carnegie Mellon University, [2]NVIDIA,
siqiouya@andrew.cmu.edu

Abstract

Simultaneous machine translation (SMT) takes streaming input utterances and incrementally produces target text. Existing SMT methods mainly use the partial utterance that has already arrived at the input and the generated hypothesis. Motivated by human interpreters' technique to forecast future words before hearing them, we propose **T**ranslation by **A**nticipating **F**uture (TAF), a method to improve translation quality while retaining low latency. Its core idea is to use a large language model (LLM) to predict future source words and opportunistically translate without introducing too much risk. We evaluate our TAF and multiple baselines of SMT on four language directions. Experiments show that TAF achieves the best translation quality-latency trade-off and outperforms the baselines by up to 5 BLEU points at the same latency (three words).

1 Introduction

Simultaneous machine translation (SMT) aims to produce translations based on partial input from the source language, enabling real-time communication across language barriers (Gu et al., 2017). Despite its potential, current SMT methods often struggle to maintain high translation quality while achieving low latency. As shown in Figure 1, the translation quality of the previous state-of-the-art method SM2 (Yu et al., 2024) drops quickly as the latency decreases. The major source of such quality drop is insufficient source information.

To deal with insufficient information, human interpreters develop techniques to anticipate future source input to reduce interpretation lag (Seeber, 2001). Human interpreters often predict upcoming words, such as nouns and verbs, based on context, language structure, prior knowledge, familiarity with the topic, etc. Most existing SMT models ignore this technique or only implicitly use it (Ma et al., 2019, 2020b; Zhang and Feng, 2022; Miao et al., 2021; Guo et al., 2024).

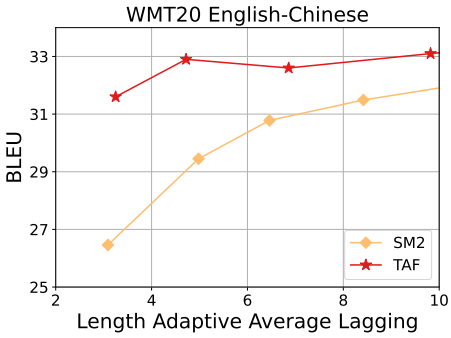

Figure 1: The quality-latency trade-off of our method TAF and the previous state-of-the-art method SM2 on WMT20 English-Chinese direction. The translation quality of SM2 drops quickly as latency goes down while TAF maintains a good translation quality even at the latency of 3 words.

A large language model (LLM) can predict the continuation of a source sentence given its prefix, mimicking this human-like anticipation (Brown et al., 2020; Touvron et al., 2023). However, perfectly predicting the continuation of a sentence is challenging due to the versatility of human language. For instance, given the prefix "The cat is chasing", various completions like "a mouse", "a bird", or "a dog" are all plausible. This variability makes it essential to develop a mechanism that ensures the translation remains consistent with the actual source when relying on the predictions.

In this work, we design a novel SMT policy, **T**ranslation by **A**nticipating **F**uture (TAF), that achieves high-quality translation at an extremely low latency. TAF utilizes an LLM to predict multiple possible continuations of the source input and translates each one with an MT model. It then employs a majority voting mechanism to select the prefix agreed upon by most candidates, ensuring consistency with source input. TAF works on any combinations of pretrained MT models and LLMs without the need for further finetuning and

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5547–5557
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

can be further generalized to existing SMT policies. Experimental results demonstrate that TAF consistently improves quality-latency trade-off over existing methods by up to 5 BLEU score at a latency of 3 words on four language directions. We conduct an in-depth analysis of the behavior of TAF to figure out the impact of LLM predictions. Finally, we find that providing the LLM with longer context further reduces the latency without sacrificing the translation quality.

2 Related Works

Recent advances in simultaneous machine translation majorly focus on its policy, either rule-based or learned adaptive. Rule-based policies include Wait-K and its variants (Ma et al., 2019; Zeng et al., 2021; Elbayad et al., 2020), Local Agreement (LA) (Liu et al., 2020a), Hold-N (Liu et al., 2020b), RALCP (Wang et al., 2024) etc. Wait-K waits for K tokens at the beginning and then alternates between Read and Write actions. LA generates a full hypothesis at each step and writes the longest common prefix (LCP) of recent hypotheses. Hold-N removes the last N tokens of the full hypothesis and writes the rest. RALCP generates multiple hypotheses with beam search and outputs a common prefix that most hypotheses agree upon.

Though rule-based policies are easy to implement, learned adaptive policies demonstrate better quality-latency trade-offs. Ma et al. (2020b) proposed monotonic multihead attention to model the policy. Miao et al. (2021) developed a generative framework with a latent variable to make decisions. Zhang and Feng (2022) quantified the information weight transported from source to target and made decisions based on the amount of information received. Zhang and Feng (2023) modeled the simultaneous translation process as a hidden Markov model and optimized the likelihood of target sequence over multiple steps. Guo et al. (2024) further bridged the gap between SMT models and offline MT models. Yu et al. (2024) is the state-of-the-art adaptive policy that resolves the insufficient exploration issue of prior works by individually optimizing each source-target state.

These learned adaptive policies implicitly model the future source input through prefix-to-prefix training. Yin et al. (2024) proposes to use a language model to explicitly predict one more source token and rewrite the translation if the prediction is wrong. However, rewriting makes it non-monotonic which limits its application to speech-to-speech scenarios. TAF explicitly models the future source with an LLM so that it is not restricted by the limited MT training data, and maintains the best translation quality at a low latency without rewriting using majority vote. Also, TAF requires no model training and works on any combination of MT models and LLMs, thus easy to implement.

3 Method

3.1 Problem Formulation

Define $\boldsymbol{x}_{1:j} = (x_1, \cdots, x_j)$ as partial text input and $\boldsymbol{y}_{1:i} = (y_1, \cdots, y_i)$ as partial hypothesis that is already generated. Define delay g_i as the number of source tokens read when generating y_i. Let $\pi(\boldsymbol{x}_{1:j}, \boldsymbol{y}_{1:i}) \in [0, 1]$ be the policy given $\boldsymbol{x}_{1:j}$ and $\boldsymbol{y}_{1:i}$, where $\pi(\boldsymbol{x}_{1:j}, \boldsymbol{y}_{1:i})$ is the probability to write $y_{i+1} \sim P_{MT}(y_{i+1} \mid \boldsymbol{x}_j, \boldsymbol{y}_i)$ with $g_{i+1} = j$, and $1 - \pi(\boldsymbol{x}_{1:j}, \boldsymbol{y}_{1:i})$ is the probability to read x_{j+1}. The complete simultaneous translation process can be formulated as follows,

$$P(\boldsymbol{y}, \boldsymbol{g}|\boldsymbol{x}) = \prod_{i=1}^{|\boldsymbol{y}|} P_{MT}(y_i|\boldsymbol{x}_{1:g_i}, \boldsymbol{y}_{1:i-1}) \times \pi(\boldsymbol{x}_{1:g_i}, \boldsymbol{y}_{1:i-1}) \prod_{j=g_{i-1}}^{g_i-1} (1 - \pi(\boldsymbol{x}_j, \boldsymbol{y}_{i-1})). \quad (1)$$

Once the translation is finished, we evaluate its quality and latency, respectively. The quality is assessed by comparing $\boldsymbol{y}$ to the ground-truth $\boldsymbol{y}^*$ using a metric $Q(\boldsymbol{y}, \boldsymbol{y}^*)$. The latency is assessed using delay $\boldsymbol{g} = (g_1, g_2, \cdots)$ together with $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{y}^*$ using a latency function $L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}^*, \boldsymbol{g})$.

3.2 Translation by Anticipating Future

Define $P^*_{LM}(x_{j+1} \mid \boldsymbol{x}_{1:j})$ as the ground-truth distribution of input, i.e., the oracle language model. Let $\boldsymbol{x}_{1:j}$ and $\boldsymbol{y}_{1:i}$ be the partial input and the generated hypothesis. An oracle simultaneous translation model P^*_{MT} generates translation y_{i+1} as if it knows the ground-truth input distribution, i.e.,

$$P^*_{MT}(y_{i+1} \mid \boldsymbol{x}_{1:j}, \boldsymbol{y}_{1:i}) = \mathbb{E}_{\boldsymbol{x}_{j+1:} \sim P^*_{LM}}[P^*_{MT}(y_{i+1} \mid \boldsymbol{x}_{1:j}, \boldsymbol{x}_{j+1:}, \boldsymbol{y}_{1:i})] \quad (2)$$

where $\boldsymbol{x}_{j+1:}$ is sampled from the oracle language model $P^*_{LM}(\cdot \mid \boldsymbol{x}_{1:j})$.

One approach to approximating such an oracle model is to train the output distribution given partial input to be close to that given full input. However, given the limited MT training data compared

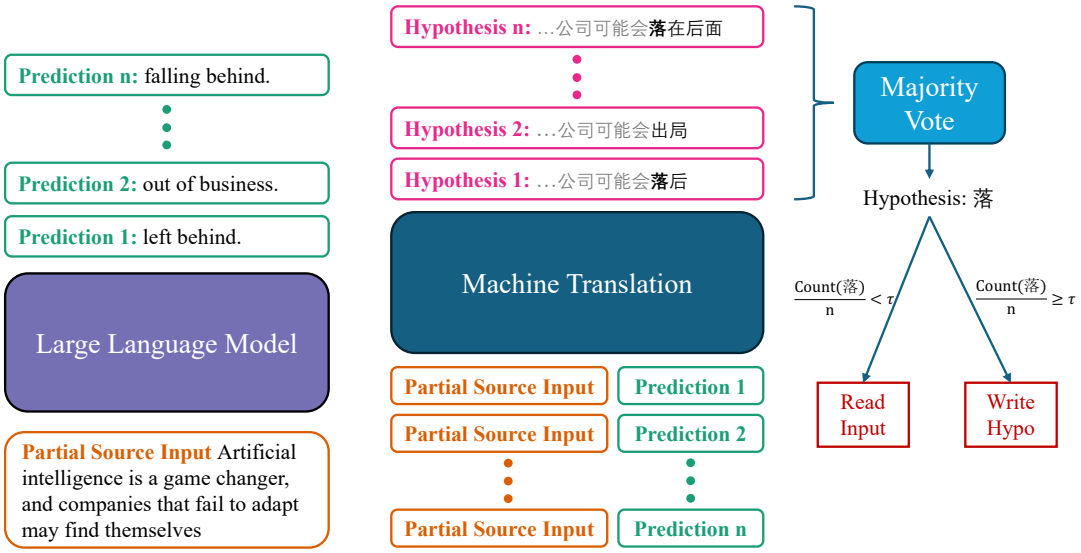

Figure 2: An overview of TAF. TAF utilizes a large language model to predict multiple possible future continuations based on partial source input, and each continuation is translated using a machine translation model. Finally, TAF applies a majority voting mechanism to select the most agreed-upon hypothesis. The system commits to the translation if the frequency of the selected hypothesis exceeds a threshold τ.

with that used for language model pre-training, it is hard for the MT model to translate while anticipating future input.

We propose to separate the translation and the anticipation. The MT model focuses only on the translation, while the language model handles the anticipation. The overview of our method is shown in Figure 2. Since we cannot access the ground-truth language model P^*_{LM}, we approximate it with a pre-trained language model P_{LM}. Then we sample n continuations of length l from P_{LM},

$$\boldsymbol{x}^1_{j+1:j+l}, \cdots, \boldsymbol{x}^n_{j+1:j+l} \tag{3}$$

where $x^t_{j+r} \sim P_{LM}(\ | \ \boldsymbol{x}_{1:j}, \boldsymbol{x}^t_{j+1:j+r-1})$ for $t \in [1, n]$ and $r \in [1, l]$. Note that we do not need to sample infinitely long continuations here because distant future source inputs will less likely affect the next token of the hypothesis.

Once we have the sampled continuations, we concatenate the received source input with the continuations and obtain the output distribution of the MT model for each of them

$$P^t_{MT} = P_{MT}(\cdot \mid \boldsymbol{x}_{1:j}, \boldsymbol{x}^t_{j+1:j+l}, \boldsymbol{y}_{1:i}). \tag{4}$$

Finally, we aggregate the n distributions $P^1_{MT}, \cdots, P^n_{MT}$ using an aggregation function f and obtain the translation y_{i+1} at this step,

$$y_{i+1} = f(P^1_{MT}, \cdots, P^n_{MT}). \tag{5}$$

Inspired by RALCP (Wang et al., 2024), let $h^t = \arg\max P^t_{MT}$ and we design the aggregation function f to be

$$f(P^1_{MT}, \cdots, P^n_{MT}) = \text{Majority}(h^t_1, \cdots, h^t_n) \tag{6}$$

where Majority() outputs the most common one of all inputs.

3.3 Policy

The policy of TAF is also a function of the output distributions,

$$\pi(\boldsymbol{x}_{1:j}, \boldsymbol{y}_{1:i}) = \frac{1}{n}\text{Count}(f(P^1_{MT}, \cdots, P^n_{MT})) \tag{7}$$

where $\text{Count}(f(P^1_{MT}, \cdots, P^n_{MT}))$ is the number of occurrences of the most common output. Intuitively, if $P^1_{MT}, \cdots, P^n_{MT}$ are vastly different from each other, then it is unlikely there will be a definite output at this step, thus we should choose to read more input. Otherwise, if most distributions are close to each other, then we are confident there will be a definite output and should output the one with which most distributions agree.

During inference, we select a threshold $\tau \in [0, 1]$ and decide to write if $\pi(\boldsymbol{x}_{1:j}, \boldsymbol{y}_{1:i}) \geq \tau$ and read otherwise. We can then obtain a quality-latency trade-off by adjusting this threshold.

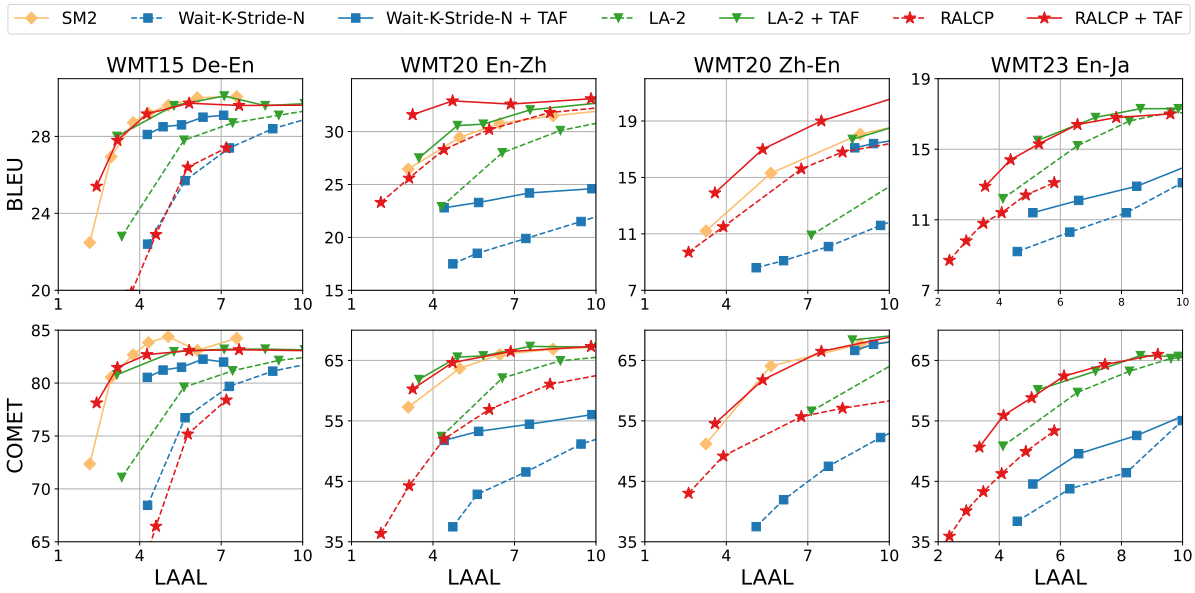

Figure 3: The quality-latency trade-off of TAF and other baselines. The quality is evaluated with both BLEU and COMET and the latency is evaluated with length adaptive average lagging (LAAL). TAF consistently improves existing policies and RALCP + TAF achieves the best performance across four language directions, with at most 5 BLEU scores improvement at a latency of 3 words in En-Zh direction.

We can also use other existing policies. We only need to switch the policy function π to those existing ones. For example, when using Wait-K policy (Ma et al., 2019), we wait for K source tokens at the beginning, then generate one token at each step using Equation 3-6.

4 Experiment Setups

4.1 Datasets

We use the following datasets to train the offline MT model and evaluate different systems.

De-En We use WMT15 as the training set which contains 4.5M sentence pairs, newstest2013 as the validation set, and newstest2015 as the test set.

En-Zh/Zh-En We use WMT20 as the training set which contains around 40M sentence pairs, newstest2019 as the validation set, and newstest2020 as the test set.

En-Ja We use WMT23 as the training set which contains around 29M sentence pairs, newstest2020 as the validation set, and newstest2023 as the test set.

4.2 System Settings

TAF We default use Llama 2 7B base (Touvron et al., 2023) as the language model for future prediction. We sample $n = 10$ future continuations with top-k sampling using $k = 10$. The maximum length of each continuation is $l = 10$. We sweep the threshold τ from 0.5 to 1.0 with step size 0.1. We use Transformer-Big (Vaswani et al., 2017) as the architecture of our offline machine translation model for all language directions. The training details are reported in Appendix A.

Wait-K-Stride-N (Zeng et al., 2021) waits K words at the beginning and then alternate between generating N words and reading one more word. We enumerate K in $[3, 6, 9, 12, 15]$. $N = 1$ for De-En/Zh-En and $N = 3$ for En-Zh/En-Ja.

Local Agreement N (LA-N) (Liu et al., 2020a) generates a hypothesis after reading each word and outputs the longest common prefix of the last N hypotheses. We vary the source segment size (the number of tokens read at each step) from 1 to 5.

RALCP (Wang et al., 2024) outputs a relaxed longest common prefix (LCP) of the beam search candidates after reading each word and finishing the beam search generation. Unlike LCP, where all candidates agree with the prefix, the relaxed prefix is the prefix that at least a fraction of candidates agree with. We vary the fraction from 0.1 to 1.0 with step size 0.1. The beam width is 40 for De-En/En-Zh and 10 for Zh-En/En-Ja.

Since TAF is compatible with existing policies, we also evaluate TAF with the above three policies.

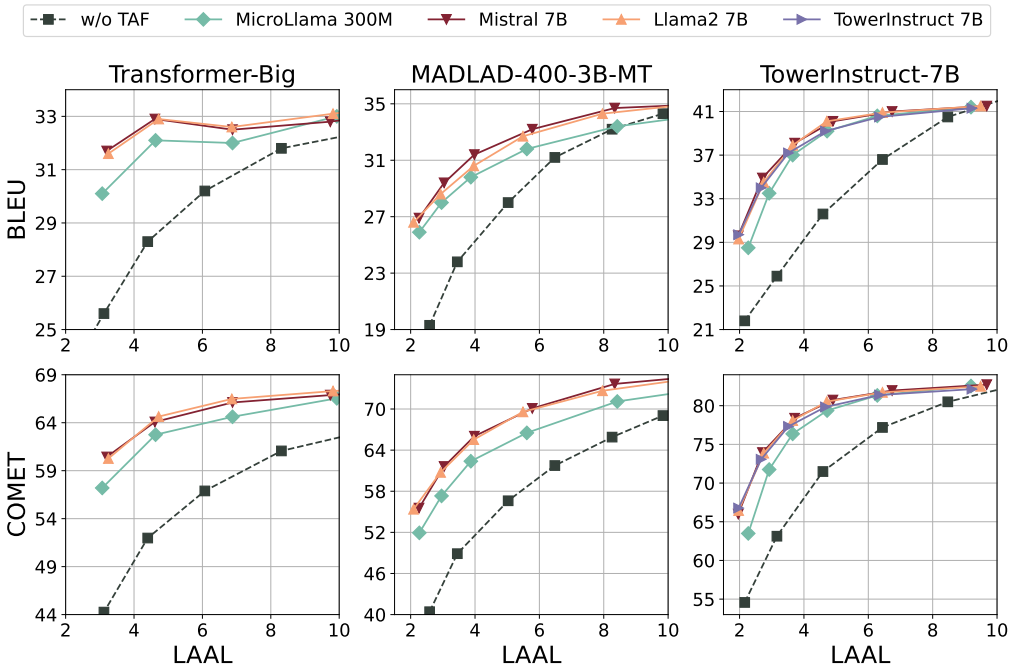

Figure 4: The quality-latency trade-off of TAF with different MT models and LLMs combinations. We also include the base RALCP (w/o TAF) results as a reference. TAF is universally effective on all combinations with at least 5 BLEU score improvement at a latency of 3 words.

Note that applying TAF on top of RALCP is equivalent to the policy mentioned in Equation 7, and we make sure the product of the number of candidates n and the beam width is equal to the beam width used in RALCP to have fair comparison.

SM2 Additionally, we compare our method with the state-of-the-art learned adaptive SMT method SM2 (Yu et al., 2024). SM2 individually optimizes the decision at each step and uses prefix sampling to ensure sufficient exploration during training. For the SM2 baseline, we follow the instructions in its original paper to obtain models of similar size to our offline model.

4.3 Evaluation Metrics

We use SimulEval (Ma et al., 2020a) to evaluate our method and other baselines. We evaluate the quality of translation by comparing the hypothesis with the ground-truth translation using BLEU (Papineni et al., 2002) and COMET (Guerreiro et al., 2024)[1]. The latency is evaluated with Length Adaptive Average Lagging (LAAL) (Papi et al., 2022). Note that we treat the word (En, De) or character (Zh, Ja) as the unit during evaluation instead of the BPE token, following the practice of recent IWSLT competitions (Ahmad et al., 2024). The latency calculated with the word or character is more intelligible than that computed with the BPE token and also enables fair comparison of models with different tokenizations.

[1] https://huggingface.co/Unbabel/XCOMET-XXL

5 Results

5.1 Best Translation Quality at an Extremely Low Latency Across Language Directions

We evaluate whether TAF improves the quality-latency trade-off over existing policies and compare it with the state-of-the-art learned model SM2. As shown in Figure 3, TAF consistently improves Wait-K-Stride-N, LA-2, and RALCP on all language directions for at least 6 BLEU scores when the latency is around 2 words. Among them, RALCP with TAF is showing the best results. It is competitive with SM2 in the De-En direction and outperforms SM2 in the En-Zh and Zh-En directions with at most 5 BLEU scores at the latency of 3 words. COMET shows similar results as BLEU. These results demonstrate that TAF achieves the state-of-the-art translation quality at an extremely low latency across different language directions.

MT Model	Size	w/o TAF	w/ TAF
Transformer-Big	0.2B	72ms	1067ms
MADLAD-400	3B	383ms	1339ms
TowerInstruct	7B	905ms	1506ms

Table 1: The average wall-clock time per step with and without TAF on different MT backbones. LLM is fixed to 7B. As the size of the MT backbone increases, the relative additional overhead introduced by LLM prediction decreases.

5.2 Generalizable to Existing Pretrained MT models and LLMs

TAF does not require sophisticated finetuning for simultaneous translation, making it easily generalizable to other pre-trained MT models and LLMs. Here we show that TAF is effective across different combinations of such models.

For MT models, we choose MADLAD-400-3B-MT (Kudugunta et al., 2023) and TowerInstruct 7B (Alves et al., 2024), as the former is a typical encoder-decoder translation model and the latter is a typical decoder-only translation model. For language models, we examine MicroLlama-300M[2] and Mistral-7B-v0.3 (Jiang et al., 2023) to find out whether TAF works for smaller LMs and other LLMs of similar size. Besides, since TowerInstruct-7B itself is an LLM, we also include the result using it for both prediction and translation. We conduct experiments on WMT20 En-Zh with RALCP+TAF policy since it performs the best, as shown previously.

Results are shown in Figure 4. TAF is universally effective on all combinations of MT models and LLMs. At a latency of 3 words, TAF improves more than 5 BLEU scores using Transformer-Big and MADLAD-400-3B-MT and more than 8 BLEU scores using TowerInstruct-7B. Also, we observe that using Mistral and Llama2 shows similar performance. A smaller LLM like MicroLlama leads to a performance drop of less than 2 BLEU scores but still 4 BLEU scores better than the base RALCP policy.

We also demonstrate the perplexity of LLMs on the source text in Table 3 together with their latency and quality scores. The effectiveness of TAF is closely correlated with the LLM perplexity on the source text.

[2] https://huggingface.co/keeeeenw/MicroLlama

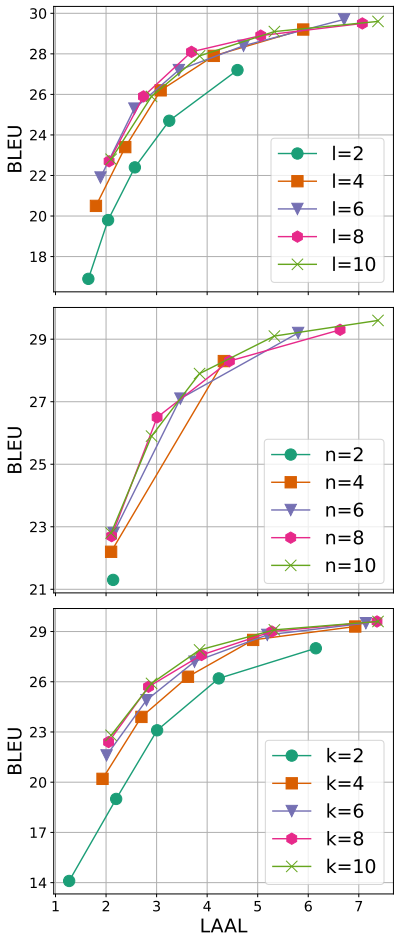

Figure 5: The quality-latency trade-off of TAF with different hyperparameters on WMT15 De-En. The improvement from TAF saturates after $l \geq 6$, $n \geq 4$ and $k \geq 8$, which means we can further reduce the computation overhead of TAF.

5.3 Computation Cost

TAF introduces additional computation overhead to conventional SMT. We report the average wall-clock time of generating a full hypothesis with RALCP+TAF versus the base RALCP policy on different combinations of MT models and a 7B LLM. We run experiments on an A6000 GPU and an AMD EPYC 9354 32-Core CPU. The results are shown in Table 1. When using a small MT model Transformer-Big (0.2B), 7B LLM introduces significant computational overhead. As the size of the MT model gets larger, the relative overhead reduces.

5.4 Sensitivity to Hyperparameters

We examine how TAF is sensitive to the choice of hyperparameters. We evaluate on De-En direction and use RALCP+TAF. We vary the number of to-

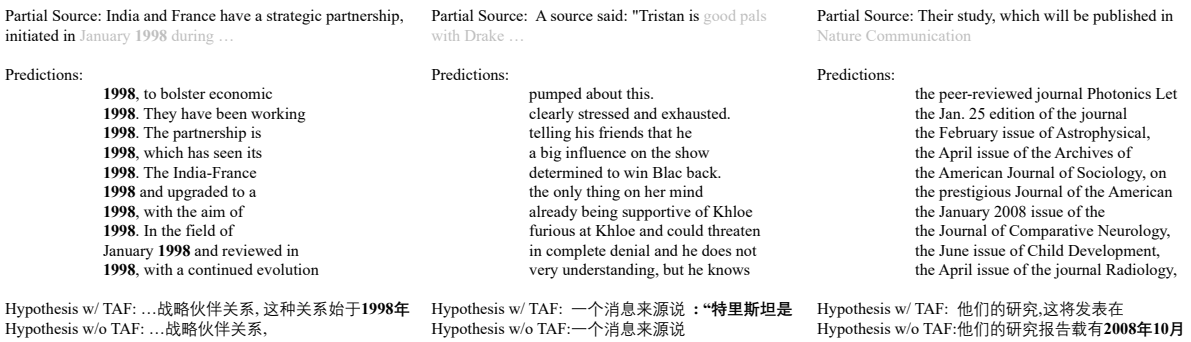

Figure 6: Cases of three typical patterns where TAF improves the quality-latency trade-off. The greyed-out text is ground-truth future source text not yet read. The first case demonstrates TAF reduces the latency with correct predictions. The second case shows TAF reduces the latency with additional context. The last case illustrates TAF reduces hallucination with additional context.

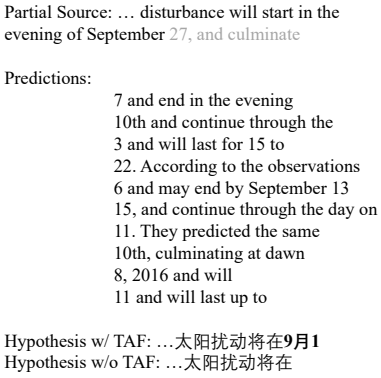

Figure 7: Case where TAF introduces additional hallucination due to LLM bias.

kens to predict l, the number of continuations n, and the k for top-k sampling from $[2, 4, 6, 8, 10]$. We sweep τ from 0.5 to 0.9 with step size 0.1 for each configuration. Note that for $n = 2$ and $\tau < 1$, there is only one data point since RALCP always follows the first of $n = 2$ translations.

Results are shown in Figure 5. As the sampling length l increases, the quality-latency trade-off improves and saturates after $l \geq 6$. This confirms that more distant source text will have a smaller impact on the hypothesis generation. As the number of candidates n increases, the trade-off also gets better but quickly saturates after $n \geq 4$, which means we can further reduce the computation cost without sacrificing too much performance. Another observation is that a larger number of candidates with the same threshold τ will have a larger latency since it requires more randomly sampled candidates to agree with each other. As k increases, we also see better results, which is consistent with Equation 2 since we want the sampled distribution to be as close to the oracle distribution as possible.

6 How TAF Impacts the Translation

We manually examine 100 instances in the En-Zh direction with the Llama2 as the LLM and MADLAD-400-3B-MT as the MT model. We compare the trajectory between RALCP with and without TAF using $\tau = 0.6$. We find four major patterns where TAF improves the quality-latency trade-off or hurts it. We show the frequency of these patterns in Table 2. If a pattern occurs multiple times in an instance, we count it once.

Reduce Latency with Correct Prediction (+) When LLM predicts the correct future words of the source, the MT model can translate those words before they appear and reduce the latency. This happens often for those entity words. Since the WMT data is from news, LLM can guess the right entity with high probability given enough context. We illustrate this with an example in Figure 6. "India and France have a strategic partnership in 1998" is already known by LLM, so the MT model can translate the year of this partnership before it is read.

Reduce Latency with Additional Context (+) When LLM prediction is not correct, it still provides additional context for the policy so that it is more confident to generate a translation of what is already read. This is illustrated in the second example in Figure 6. With LLM prediction, the MT model is confident in translating "Tristan is" into Chinese, but without future prediction, the model stays conservative and needs more input to continue the translation.

Pattern	Frequency	Δ COMET	Δ LAAL
All 100 Instances	100%	+6.78	-0.25
↓ Latency w/ Correct Prediction	44%	+10.61	-0.45
↓ Latency w/ Additional Context	53%	+5.42	-1.05
↓ Hallucination w/ Additional Context	39%	+24.44	+0.29
↑ Hallucination w/ LLM Bias	28%	-6.99	-0.31

Table 2: Statistics on the impact of TAF on hypothesis generation across different patterns in 100 manually examined instances. Δ represents the difference between the results of RALCP with and without TAF.

Prevent Hallucination with Additional Context (+) The MT model will often generate hallucinated content that does not appear in the source or generate low-quality translation with insufficient context. The prediction of LLM provides additional context to the policy to realize the possible continuations of the source and stays conservative if the current context is not enough for generation. This is illustrated in the last example in Figure 6. Without future prediction, the MT model generates hallucination (highlighted in bold) without any such information from the source. This is probably caused by bias during MT model training and can be avoided with extended context from the LLM.

Introduce Additional Hallucination (-) In certain cases, the bias of LLM also introduces additional hallucination during simultaneous translation. As shown in Figure 7, the LLM predicts "September 1" when only "September" appears in the source, but the true date of the event is "September 27". As shown in Table 2, it slightly worsens translation quality in some cases. However, when considering both the reduction and the introduction of hallucinations, TAF ultimately improves overall translation quality.

7 Better Prediction with Longer Context

In real-world SMT applications, such as multilingual conferences, speeches can last minutes or hours, allowing models to leverage the full context of prior speech. However, WMT datasets are pre-segmented into sentences. When evaluating SMT on isolated sentences, LLMs cannot access the prior context of the document, making it challenging to anticipate future content accurately.

For instance, given the partial source "Regular physical activity can significantly reduce the risk of", the next phrase could refer to various diseases related to physical activity. However, if the prior context focuses on heart health, the continuation will likely be "heart disease".

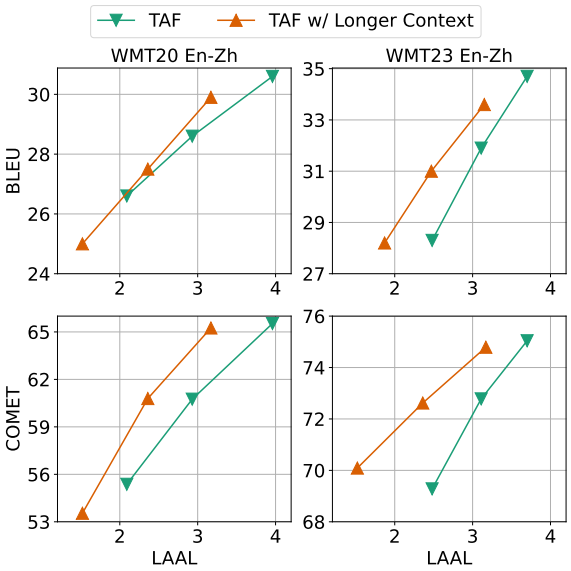

Figure 8: The quality-latency trade-off of TAF with and without longer context on WMT20 and WMT23 En-Zh test set. Longer context consistently reduces the latency of TAF by 0.5 to 1 word.

To better simulate real-world scenarios, we prepend the current partial source with previous sentences from the same document, allowing the LLM to predict based on this extended context. Before passing these predictions to the MT model, we remove the past sentences to ensure that translation remains sentence-level. This setup guarantees that the LLM's context is the only variable being tested.

Results are shown in Figure 8. TAF, when using a longer context, consistently reduces latency by 0.5 to 1 word while maintaining comparable translation quality on both WMT20 and WMT23 En-Zh. Since WMT23 test data is less likely to have appeared in Llama2's training set, the improvement from TAF may not be solely due to triggering the LLM's memory. Instead, the longer context could

enable the LLM to make more informed predictions about future content.

8 Conclusion

We propose TAF , a novel SMT method that translates by anticipating future source input. Experiments on four language directions show that TAF achieves state-of-the-art quality-latency tradeoff and is universally effective on different combinations of pretrained MT models and LLMs. Our manual analysis reveals how TAF impacts the translation output. Finally, we show that TAF can be further improved with a longer context.

Limitations

Apart from the additional computation cost mentioned in Section 5.3 and the hallucination caused by LLM bias in Section 6, we have yet to explore other choices of the aggregation function. It can be simply a mean pooling function or a more advanced function that takes the semantic meaning into account. Besides, our experiments focus on X-En and En-X directions. X-X directions are not covered yet. Also, our experiments are only on text-to-text translation. The major obstacle to migrating TAF to speech-to-text translation is that predicting continuous future audio signals is very hard. A possible solution could be a cascade speech-to-text model with TAF applied to the transcribed speech.

References

Ibrahim Said Ahmad, Antonios Anastasopoulos, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, William Chen, Qianqian Dong, Marcello Federico, Barry Haddow, Dávid Javorský, Mateusz Krubiński, Tsz Kim Lam, Xutai Ma, Prashant Mathur, Evgeny Matusov, Chandresh Maurya, John McCrae, Kenton Murray, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Atul Kr. Ojha, John Ortega, Sara Papi, Peter Polák, Adam Pospíšil, Pavel Pecina, Elizabeth Salesky, Nivedita Sethiya, Balaram Sarkar, Jiatong Shi, Claytone Sikasote, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Brian Thompson, Alex Waibel, Shinji Watanabe, Patrick Wilken, Petr Zemánek, and Rodolfo Zevallos. 2024. FINDINGS OF THE IWSLT 2024 EVALUATION CAMPAIGN. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 1–11, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. In *First Conference on Language Modeling*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient wait-k models for simultaneous machine translation. In *Interspeech 2020*, pages 1461–1465.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent Machine Translation Evaluation through Fine-grained Error Detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Shoutao Guo, Shaolei Zhang, and Yang Feng. 2024. Glancing future for simultaneous machine translation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11386–11390.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Sneha Kudugunta, Isaac Rayburn Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. MADLAD-400: A multilingual and document-level large audited dataset. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020a. Low-latency sequence-to-sequence speech

recognition and translation by partial hypothesis selection. In *Interspeech 2020*, pages 3620–3624.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020b. Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection. In *Interspeech 2020*, pages 3620–3624.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020b. Monotonic multihead attention. In *International Conference on Learning Representations*.

Yishu Miao, Phil Blunsom, and Lucia Specia. 2021. A generative framework for simultaneous machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6697–6706, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 12–17, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Kilian G Seeber. 2001. Intonation and anticipation in simultaneous interpreting. *Cahiers de linguistique française*, 23(1):61–97.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Minghan Wang, Jinming Zhao, Thuy-Trang Vu, Fatemeh Shiri, Ehsan Shareghi, and Gholamreza Haffari. 2024. Simultaneous machine translation with large language models. *Preprint*, arXiv:2309.06706.

Aoxiong Yin, Tianyun Zhong, Haoyuan Li, Siliang Tang, and Zhou Zhao. 2024. Language model is a branch predictor for simultaneous machine translation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*, pages 9976–9980. IEEE.

Donglei Yu, Xiaomian Kang, Yuchen Liu, Yu Zhou, and Chengqing Zong. 2024. Self-modifying state modeling for simultaneous machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9781–9795, Bangkok, Thailand. Association for Computational Linguistics.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2461–2474, Online. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022. Information-transport-based policy for simultaneous translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 992–1013, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2023. Hidden markov transformer for simultaneous machine translation. In *The Eleventh International Conference on Learning Representations*.

Model	Context	Perplexity	LAAL	COMET
MicroLlama-300m	Sentence	113.63	2.27	51.93
Llama2-7b	Sentence	12.86	2.09	55.37
Mistral-7b	Sentence	14.52	2.26	55.54
Llama2-7b	Document	5.76	2.36	60.79

Table 3: LLM perplexity correlates well with simultaneous translation performance.

A Training Details of Offline MT

We trained our NMT models (Transformer, 12×6 layers, $d_{\text{model}} = 1024$, $d_{\text{inner}} = 4096$, $n_{\text{heads}} = 16$) with Adam optimizer and inverse square root annealing (Vaswani et al., 2017) with 7.5K warmup steps and a maximum learning rate of 10^{-3}. The models were trained for a maximum of 100K steps with a dropout of 0.1 on intermediate activations and label smoothing with $\alpha = 0.2$. Our De→En models used joint BPE vocabulary of 16384 tokens and En↔Zh/Ja used separate vocabularies with the same number of tokens per language.

B LLM Perplexity on Source Text

To find out the correlation between LLM's next token prediction and the final simultaneous translation performance, we measure the perplexity of MicroLlama-300m, Llama2-7b and Mistral-7b on source sentences of WMT20 En-Zh test set. We also measure the perplexity of Llama2-7b on unsegmented source documents as in Section 7. The results are shown in Table 3.