Quantifying Information of Tokens for Simple and Flexible Simultaneous Machine Translation

DongHyun Lee, Minkyung Park, Byung-Jun Lee
Department of Artificial Intelligence
Korea University, Republic of Korea
{2022020880, swwwjkl538, byungjunlee}@korea.ac.kr

Abstract

Simultaneous Translation (ST) involves translating with only partial source inputs instead of the entire source inputs, a process that can potentially result in translation quality degradation. Previous approaches to balancing translation quality and latency have demonstrated that it is more efficient and effective to leverage an offline model with a reasonable policy. However, using an offline model also leads to a distribution shift since it is not trained with partial source inputs, and it can be improved by training an additional module that informs us when to translate. In this paper, we propose an Information Quantifier (IQ) that models source and target information to determine whether the offline model has sufficient information for translation, trained with oracle action sequences generated from the offline model. IQ, by quantifying information, helps in formulating a suitable policy for Simultaneous Translation that better generalizes and also allows us to control the trade-off between quality and latency naturally. Experiments on various language pairs show that our proposed model outperforms baselines. [1]

1 Introduction

Simultaneous Translation (ST)(Kreutzer et al., 2018; Gu et al., 2017) is a setting that employs incremental translation as the source input is being received, unlike conventional Machine Translation (MT)(Vaswani et al., 2017) which translates using full source sentences, providing a sufficient context for high-quality translation. Despite its invaluable potential in numerous real-world scenarios, ST poses a significant challenge as the translation model may not always have access to sufficient source context, particularly under low latency conditions.

In the pursuit of achieving Simultaneous Translation (ST), a multitude of methods have been proposed for the training of online models, employing either fixed policies (i.e., Wait-k) (Ma et al., 2019; Zheng et al., 2020; Elbayad et al., 2020; Zhang and Feng, 2021), or adaptive policies (Chiu and Raffel, 2018; Arivazhagan et al., 2019; Ma et al., 2020b; Zhang and Feng, 2022a, 2023). Regardless, the training of a dedicated online model for ST often requires calibration of diverse factors to control latency, such as the count of reading windows (i.e., k), and latency weight. This typically induces the training of multiple models, thereby incurring high computational costs. While it is possible to consider multiple latency regimes within a single model (Elbayad et al., 2020; Zhang and Feng, 2021), it does not account for the correlation between different latency conditions (Zhang and Feng, 2022b).

In recent research, (Papi et al., 2022) showed the effectiveness of directly deploying an offline model with a suitable decision policy for ST. Their promising results demonstrate that we can attain superior performance without having to depend on online models that are trained using incomplete inputs. Despite their promising results, it is apparent that employing the offline model directly will suffer from a distributional shift caused by the partial source sentences that were not encountered during the training time. One previous work (Alinejad et al., 2021) has alleviated it by training a policy to predict optimal translation points, we empirically found that such an approach struggles to generalize effectively when faced with unseen source sentences.

To this end, we propose Information Quantifier (IQ) which models source and target information based on the given oracle action sequences. IQ is capable of quantifying the information contained within the source/target sentences, thereby guiding READ/WRITE decisions across diverse latency

[1] Code is available at https://github.com/ku-dmlab/info_quantifier

Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL), pages 200–210
December 6–7, 2023.

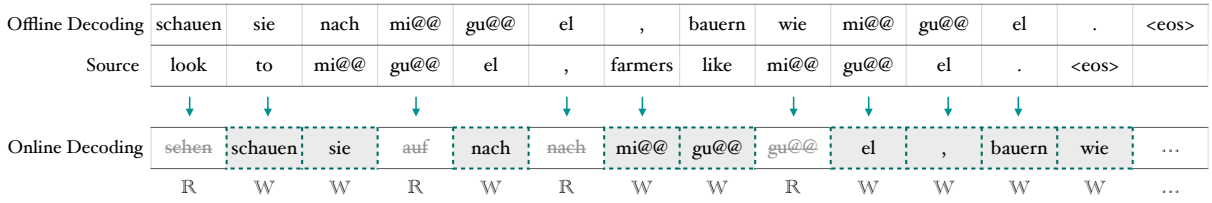


Figure 1: Example of oracle action sequences generation as suggested by SSMT (Alinejad et al., 2021). It assumes that WRITE (𝕎) is the right action to do when the decoder with partial source sentence (Online Decoding above) produces the same target token as the decoder with full source sentence (Offline Decoding above), and READ (ℝ) otherwise.

regimes by measuring the amount of excessive information in source/target sentences when compared to each other. This allows our approach to have improved generalization to mitigate distribution shift on unfamiliar source/target sentences compared to methods that directly predict actions. Through experiments across various language pairs, we demonstrate that IQ, despite its straightforward usage, delivers notable performance improvement over a number of baselines.

2 Related Work

Online models for ST Online models with a fixed policy (i.e., Wait-*k*) (Ma et al., 2019) are trained by waiting for a predefined number of *k* source tokens. Instead of training multiple *k* models (Zheng et al., 2020), strategies for training a single model for different latencies have been proposed. (Zhang and Feng, 2021) use each head in multi-head attention modules as an expert with its own *k*, while (Elbayad et al., 2020) samples *k* randomly during training. Online models with an adaptive policy employ specific signals to guide READ/WRITE decisions, thereby learning a flexible policy. For instance, (Ma et al., 2020b) incorporates (Arivazhagan et al., 2019), which predicts a Bernoulli variable to determine when to translate within a transformer by jointly learning with multi-head attention. Furthermore, (Zhang and Feng, 2022b; Zhang et al., 2022; Dong et al., 2022) learn the ST model with the module that quantifies information to grasp READ/WRITE decisions. While the latter provides a better trade-off between quality and latency than the former, its learning process is more intricate.

Offline model with decision policy Recent studies (Papi et al., 2022) demonstrate the efficiency and effectiveness of applying predefined or learned policy to an offline model for Simultaneous Speech Translation, as opposed to training online models. Predefined policies such as Wait-*k* (Ma et al., 2019), Wait-*k*-Stride-*n* (Zeng et al., 2021), SP-*n* (Shared prefix) (Nguyen et al., 2021), LA-*n* (Local Agreement) policy (Liu et al., 2020; Polák et al., 2022) can be applied to the offline model for ST. Additionally, (Papi et al., 2023) incorporates a policy that takes into account the attention weights of the most recent source tokens.

(Alinejad et al., 2021) suggested learning a policy model separately using oracle action sequences. We follow the same process to generate oracle action sequences. However, instead of training a policy to directly predict the actions, we introduce information quantification for decision policy which subsequently enhances the generalization capabilities of the model. In contrast to previous methods that quantify information (Zhang and Feng, 2022b; Zhang et al., 2022; Dong et al., 2022) based on heuristic policies such as the Wait-*k* policy or cross-attention values within the online model learning framework, our approach strategically aligns information learning with the action sequences generated by the oracle policy, which is entirely independent of the translation learning pipeline.

3 Background

Offline and online decoding We denote the source tokens as $\mathbf{x} = (x_1, \ldots, x_m) \in X$ and the generated target tokens as $\mathbf{y} = (y_1, \ldots, y_n) \in Y$. Offline decoding uses full-sentence inputs for training, with the greedy target token at a time step t defined as:

$$y_t = \arg\max_{\mathrm{y}} p(y|\mathbf{x}, y_{<t})$$

Oracle action sequences Oracle action sequences are the reference that can achieve high quality under low latency in online decoding for ST. For the parallel corpus for training, the target

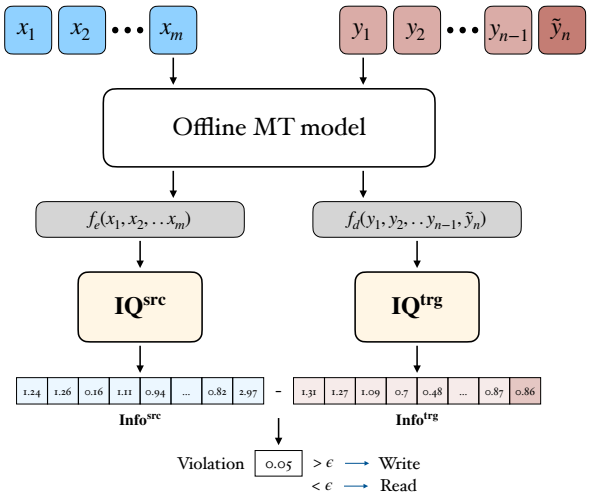


Figure 2: Overall Information Quantifier (IQ) framework. IQ networks are trained to not violate the assumptions on information of source tokens and target tokens (including the predicted candidate target token at the last). After training, the information on source tokens and partial translation information are compared to decide the next action.

sentences are given, and such action sequences can be generated in many different ways (e.g. performing a search).

As shown in Figure 1, (Alinejad et al., 2021) finds a near-optimal oracle action sequence by defining the *optimal segment*. It is the point when the target token in offline decoding (i.e., generating with complete source inputs) and the target token in online decoding (i.e., generating with incomplete source inputs) are the same. We used the same process to get oracle action sequences, primarily owing to its straightforwardness and efficiency. However, it should be noted that our proposed method can be integrated with any other oracle action sequences such as (Zheng et al., 2019b,a).

4 Propose Method

In this section, we introduce Information Quantifier (IQ), which quantifies the information in both source and target sentences to make the right READ/WRITE decisions. Based on oracle READ/WRITE action sequences of training parallel corpus (e.g., (Alinejad et al., 2021)), we train IQ with a novel training objective in the following subsections.

4.1 Quantify information

Motivated by previous studies (Zhang et al., 2022; Zhang and Feng, 2022b), we quantify the information contained in each token using a scalar value. We sum up the amount of information of tokens in a partial sentence to get the amount of information of a partial sentence. These amounts of information of source/target sentences are denoted as $\mathbf{Info}^{src} : X \mapsto \mathbb{R}$ and $\mathbf{Info}^{trg} : Y \mapsto \mathbb{R}$, respectively. We utilize the contextual token features and a feed-forward network to quantify the information contained in the source sequence $\mathbf{x} = (x_1, \ldots, x_m)$ and target sequence $\mathbf{y} = (y_1, \ldots, y_{n-1}, \tilde{y}_n)$:

$$\mathbf{Info}^{\mathbf{src}}(\mathbf{x}) = \sum_{k=1}^{m} \mathbf{IQ}^{\mathbf{src}}(f_e(x_k)) \quad (1)$$

$$\mathbf{Info}^{\mathbf{trg}}(\mathbf{y}) = \sum_{k=1}^{n} \mathbf{IQ}^{\mathbf{trg}}(f_d(y_k)) \quad (2)$$

$\mathbf{IQ}^{\mathbf{src}}$ and $\mathbf{IQ}^{\mathbf{trg}}$ stand for **Source Information Quantifier** and **Target Information Quantifier** respectively. These are the feed-forward networks that map contextual token features to the amount of information contained in the token. We use the *softplus* activation function at the end of these networks to ensure the positivity of the amount of information in each token. f_e and f_d are contextual token feature extractors from the encoder/decoder pre-trained for offline translation.

One important detail here is that, in addition to current partial source/target sentences, we also include the candidate target token $\tilde{y}_n$ that will be decoded if we perform the WRITE action for the information quantification of the target sentence. It allows the IQ model to peak into the future to make more accurate decisions.

4.2 Violation and objective

To train IQ, we introduce a novel objective based on a measure of *violation* that current IQ has on the oracle action sequences. The definition of *violation* is as follows:

$$\text{viol}(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{Info}^{\mathbf{trg}}(\mathbf{y}) - \mathbf{Info}^{\mathbf{src}}(\mathbf{x}) & \text{if READ} \\ \mathbf{Info}^{\mathbf{src}}(\mathbf{x}) - \mathbf{Info}^{\mathbf{trg}}(\mathbf{y}) & \text{if WRITE} \end{cases} \quad (3)$$

The idea behind *violation* we have assumed is as follows:

- For **READ** in action sequences, the amount of information of the target tokens should be greater than that of the source tokens (i.e., we do not have enough information in the source sentence to write).

- For **WRITE** in action sequences, the amount of information of the source tokens should exceed that of the target tokens (i.e., we do have enough information in the source sentence to write).

Based on these ideas, *violation* measures how much of these assumptions are violated. If $\text{viol}(\mathbf{x}, \mathbf{y})$ is less than zero, we can safely state that none of these assumptions are violated for current $\mathbf{x}, \mathbf{y}$. These give rise to the following objective:

$$\min \max \{\text{viol}(\mathbf{x}, \mathbf{y}), 0\}, \tag{4}$$

which is designed to only penalize the positive *violation* and ignore it if it is negative. One particular loss function to achieve it would be:

$$\mathcal{L}_{viol} = \max(\text{viol}(\mathbf{x}, \mathbf{y}), 0)^2. \tag{5}$$

However, solely using $\mathcal{L}_{viol}$ can easily lead to the trivial degenerate solution $\mathbf{Info}^{src}(\mathbf{x}) = \mathbf{Info}^{trg}(\mathbf{y}) = 0$ for all $\mathbf{x}, \mathbf{y}$, which gives $\mathcal{L}_{viol} = 0$. Such a solution is obviously not a desired outcome. To address this issue, we introduce an auxiliary objective to the information quantifier that benefits non-zero quantification:

$$\mathcal{L}_{info} = \|\mathbf{Info}^{\mathbf{trg}}(\mathbf{y}) + \mathbf{Info}^{\mathbf{src}}(\mathbf{x}) - \zeta\|^2, \tag{6}$$

where ζ represents the total information. We use the simple heuristics to set $\zeta = n + m$, which tries to equate the total sum of the amount of information to the total length of the source and target sequences. Note that, as we use contextual feature vectors as input to IQs, this auxiliary objective does not harm the expressivity of our framework.

Based on the above, we optimize IQs based on the combination of two losses:

$$\mathcal{L} = \mathcal{L}_{viol} + \alpha \mathcal{L}_{info} \tag{7}$$

where α is a hyperparameter to be tuned.

4.3 Inference

At a test time, based on IQs learned, we need to decide whether to READ or WRITE. As we trained IQs to minimally violate the assumptions, we can expect them to follow the assumptions during the test time if they generalize well. Consequently, the main idea is to follow the assumptions to perform a ST:

- Choose **READ** if the amount of information of the target tokens is larger than that of the source tokens.

Algorithm 1 Inference with IQs

Input: source tokens $\mathbf{x}$, threshold ϵ
Output: translation $\mathbf{y}$
Init: source index $i = 1$, target index $j = 0$
while $y_{j-1} \neq$ <EOS> **do**
  Predict the candidate translation $\tilde{y}_{j+1}$
  Compute $\mathbf{Info}^{\mathbf{src}} = \mathbf{Info}^{\mathbf{src}}(x_1, ..., x_i)$
  Compute $\mathbf{Info}^{\mathbf{trg}} = \mathbf{Info}^{\mathbf{trg}}(y_1, ..., \tilde{y}_{j+1})$
  if $\mathbf{Info}^{\mathbf{src}} - \mathbf{Info}^{\mathbf{trg}} \geq \epsilon$ **then**
    WRITE, j ← j+1
  else
    READ, i ← i+1
  end if
end while

- Choose **WRITE** if the amount of information of the source tokens is larger than that of the target tokens.

In practice, there is a need to control a trade-off between quality and latency. One major advantage of the proposed framework is that we can simply adjust it after training IQs. We can additionally adopt a threshold ϵ such that the **WRITE** action is performed when $\mathbf{Info}^{\mathbf{src}}(\mathbf{x})$ is larger than $\mathbf{Info}^{\mathbf{trg}}(\mathbf{y}) + \epsilon$, preventing the translator to write until the additional information ϵ is provided. The detailed algorithm is illustrated in 1.

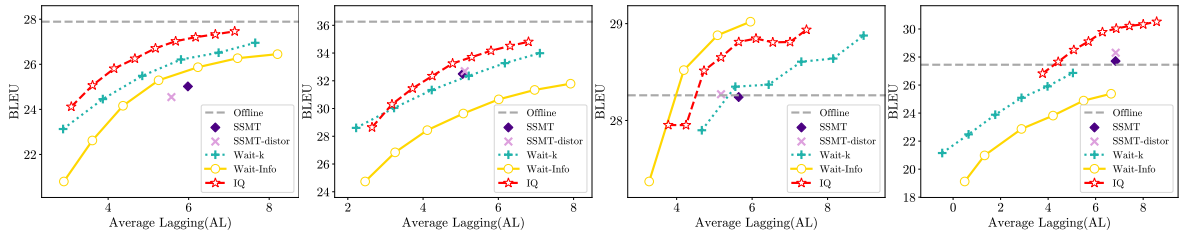
5 Experiments

5.1 Datasets

We evaluated our method on IWSLT14 (Cettolo et al., 2013) De → En, En → De, and IWSLT15 (Cettolo et al., 2015) Vi → En, En → Vi datasets.

For IWSLT14 De-En pairs, we applied Byte Pair Encoding (BPE) (Sennrich et al., 2016) to create subword vocabularies with 8.8K German and 6.6K English tokens. We used 160K and 7K sentences for the training and validation sets respectively. The test set included 6.7K sentences from `dev2020` and `tst2010-2013`.

For the IWSLT15 Vi-En pairs, we followed the settings outlined in (Luong and Manning, 2015). We utilized pre-tokenized sentence datasets with vocabularies of 17K for English and 7.7K for Vietnamese. We maintained casing for words and replaced words occurring less frequently than 5 times with <*UNK*>, as done in (Luong and Manning, 2015). The training set consisted of 133K sentences, with 1.5K sentences from `tst2012` serving as the validation set, and 1.2K sentences from



a) En→De b) De→En c) En→Vi d) Vi→En

Figure 3: Comparison with related methods: we perform evaluations across 4 language pairs, comparing the performance of the IQ against the offline model with the Wait-*k* policy, SSMT, and Wait-Info.

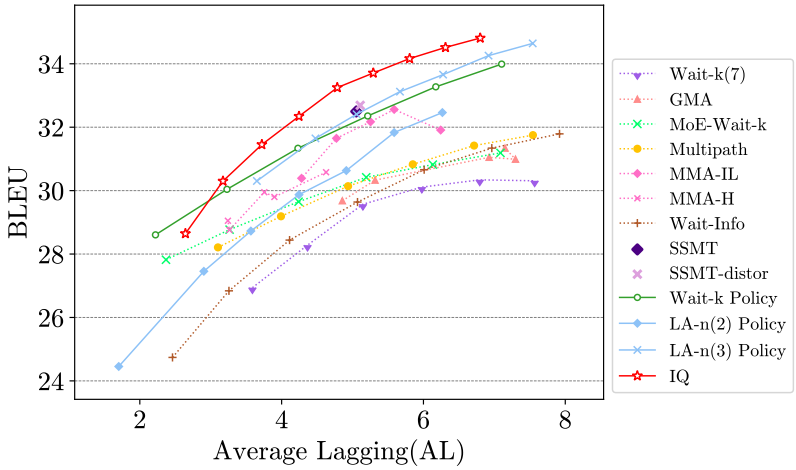


Figure 4: Evaluation against diverse algorithms: assessing online and offline models with decision policies on Simultaneous Translation (ST) results for the IWSLT14 De → En language pair. The dashed line represents the online model, while the solid line denotes the offline model with policy. The pre-trained offline model used in some of the algorithms attains a BLEU score of 36.25 when full source sentences are given.

`tst2013` used as the test set to train our model.

5.2 Baseline settings

We conducted experiments with the following baselines. If a hyperlink is accompanied by a baseline below, it implies that we used the implementation and hyperparameters of the linked implementation.

Offline Model We adopted the conventional transformer architecture model (Vaswani et al., 2017) as the offline MT model with greedy decoding. For training the policy model, we use the same offline model for each language pair, adapted from the Fairseq[2] (Ott et al., 2019) Library (`transformer_iwslt_de_en` architecture). We retained all the original hyperparameters as per the Fairseq settings, without any changes. **Offline Model with Wait-*k* Policy** Offline model with Wait-*k* policy (Ma et al., 2019) which waits for a fixed number of source tokens to be fed into the pre-trained offline model.

Offline Model with LA-*n* Policy Offline model with the local agreement (LA-*n*) policy (Liu et al., 2020), which emits the agreeing prefix tokens of the consecutive tokens. After the model receives the number of *n* source tokens, the LA-*n* policy determines the longest common prefix of the hypothesis tokens from the *n* consecutive source tokens.

Wait-*k* Model An online model is trained with a dedicated k_{train} and evaluated with k_{test} (Ma et al., 2019) to accommodate different latency regimes.

GMA[3] An online model employs a gaussian prior to learn the alignments within the attention mechanism that is used to determine READ/WRITE action (Zhang and Feng, 2022a).

MMA An online model that uses the prediction of a Bernoulli variable to determine READ/WRITE actions within a Transformer (Ma et al., 2020b).

MoE Wait-*k*[4] An online model that employs each head in the multi-head attention as an expert, which each one processing its own *k* (Zhang and Feng, 2021).

[2]https://github.com/facebookresearch/fairseq

[3]https://github.com/ictnlp/GMA

[4]https://github.com/ictnlp/MoE-Waitk

Source		ich	sehe	den	d@@	al@@	ma@@	tin@@	er	.									
Reference		then	i	see	the	d@@	al@@	ma@@	ti@@	an	.								
SSMT	**Input**	ich		sehe		den	d@@		al@@	ma@@	tin@@	er	.						
	Output		i		see			the						d@@	al@@	ma@@	tin@@	er	.
IQ(Ours)	**Input**	ich	sehe		den		d@@	al@@		ma@@		tin@@			er		.		
	Output		i	i	see	see	it	the	the	d@@	d@@	al@@	al@@	ma@@	tin@@	tin@@	er	er	.
	Src info		1.24	2.5	2.5	2.66	2.66	3.37	4.31	4.31	5.11	5.11	6.04	6.04	6.04	6.86	6.86	9.83	9.83
	Trg info		1.31	1.31	2.58	2.58	3.8	3.67	3.67	4.37	4.37	5.4	5.4	5.88	6.64	6.64	8.05	8.92	9.78
	Viol		-0.06	1.19	-0.07	0.07	-1.14	-0.29	0.64	-0.06	0.73	-0.29	0.64	0.166	-0.59	0.223	-1.19	0.91	0.05

WRITE Trg Info Degration

Figure 5: The table illustrates the different approaches IQ and SSMT take in ST processes. READ/WRITE decisions of IQ are guided by the `violation` value, offering control over latency. Notably, the portion marked red indicates situations where higher target information leads to READ when the current hypothesis lacks information for target token emission. The information for 'it' drops to 3.67 upon decoding 'the'.

Multipath[5] An online model is trained through random sampling of k, enabling it to operate under different latency conditions with just a single model (Elbayad et al., 2020).

Wait-Info[6] An online model used the attention distribution to measure the information contained in each token in an unsupervised manner (Zhang et al., 2022).

SSMT[7] A policy model is trained with oracle action sequences generated from the offline model in a supervised manner to predict READ/WRITE decisions directly (Alinejad et al., 2021). SSMT-distor introduces distortion by swapping READ to WRITE or vice versa if both source and target tokens are not the <*EOS*> token in the generated action sequence, which enhances model robustness. We used the same offline model as IQ to generate oracle action sequences.

IQ Proposed framework in Sec. 4. As illustrated in Figure 2, we adopted fully connected neural networks with 3 hidden layers for both $\mathbf{IQ}^{\mathbf{src}}$ and $\mathbf{IQ}^{\mathbf{trg}}$ to learn the source and target information. The dimensions of the layers were set to 512 to match the dimensions of the Transformer. In the encoder and decoder of the offline model, the last hidden states of the source and target are fed into the $\mathbf{IQ}^{\mathbf{src}}$ and $\mathbf{IQ}^{\mathbf{trg}}$, respectively.

5.3 Main results

In this section, we evaluate the effectiveness of our approaches. We employ SimulEval (Ma et al., 2020a) to provide accurate reporting of Corpus-BLEU, via SacreBLEU (Post, 2018), for translation quality and Average Lagging (AL) (Ma et al., 2019) for latency. All the performance metrics reported herein are derived using greedy decoding.

Comparison to related algorithms Figure 3 shows the performance for the En ↔ De, En ↔ Vi pairs when evaluated with our model against the closely related previous works: SSMT that is trained with the same oracle action sequences, and Wait-Info that also tries to capture the amount of information in each token. These results show that IQ successfully improves from other related algorithms, outperforming all the other algorithms except for En → Vi pair. While we have only varied the threshold ϵ from 0 to 4, increasing in steps of 0.5, it is also possible to easily adjust latency further by setting ϵ below 0 or above 4.

Comparison to diverse baselines We also compare to various online models, namely, Wait-k, GMA, MoE Wait-k, Multipath, MMA, and Wait-Info, represented by dashed lines in Figure 4. For offline models with predefined policy, we select the Wait-k and LA-n policies represented by dashed lines, along with the learned policy from SSMT. Our proposed framework (IQ) outperforms all baselines in achieving the most advantageous quality-latency trade-off.

It can be observed that baselines employing policy on offline models tend to exceed online models in performance. These results support the premise that an offline model, trained with complete sentences, acquires a more comprehensive context, thereby enhancing ST capabilities. In contrast, an online model may suffer performance setbacks due to inadequate information learned from incomplete sentences, as indicated by (Papi et al., 2022).

[5] https://github.com/elbayadm/attn2d
[6] https://github.com/ictnlp/Wait-info
[7] https://github.com/sfu-natlang/Supervised-Simultaneous-MT

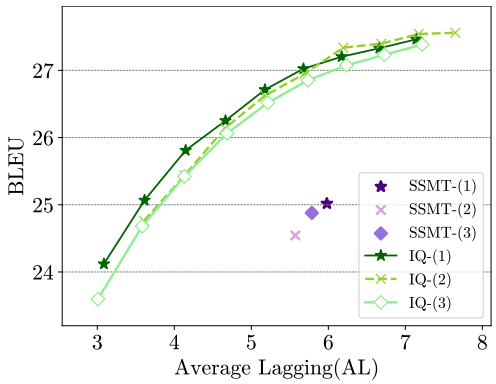


Figure 6: Performance comparison with different data generation strategies on En → De.

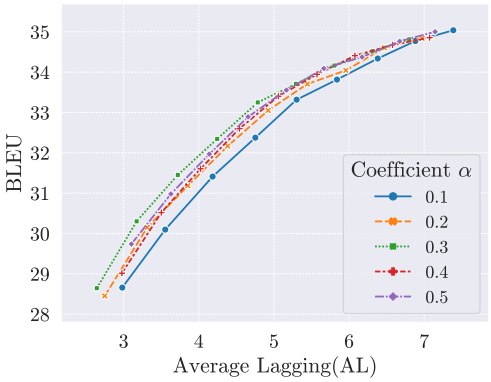


Figure 7: Performance comparison with varying parameter α on De → En.

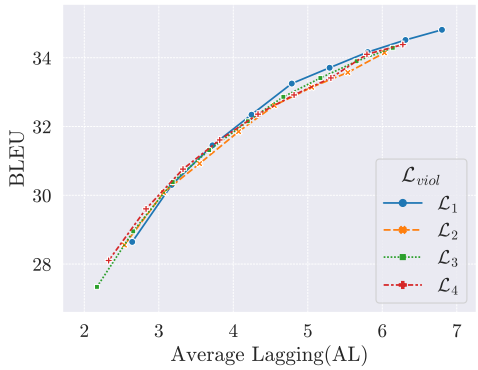


Figure 8: Performance comparison with varying loss functions on De → En.

6 Analysis

We conducted additional experiments and analyses to better understand how our method works and show improvements. All analyses are based on either the IWSLT14 De→En test set or IWSLT14 En→De test set.

6.1 Impact of dataset generation strategies

We examined several different strategies for generating oracle action sequences:

1) Base The main strategy used in main experiments. It does not generate more action sequences after reading all source tokens.

2) Distortion The data distortion method from SSMT that detailed in Sec 5.3.

3) Complete Strategy including all the WRITE decisions after reading all source tokens.

The results are shown in Figure 6. Overall, our proposed IQ framework shows robust performance over a set of different oracle action generation strategies. It can be noted that the distortion strategy additionally proposed by (Alinejad et al., 2021) is unnecessary for our framework. Excluding a series of WRITE actions at the end slightly improves the performance of our framework, presumably due to the removal of unnecessary regularization from additional $\mathcal{L}_{info}$.

6.2 Differences across various α

To demonstrate the effects of varying the coefficient α, we conducted experiments by varying α from 0.1 to 0.5 in steps of 0.1. As can be observed in Figure 7, at lower latency, a coefficient of 0.3 delivers the best performance, while at higher latency, the performances appear to be similar. This also underscores that our method exhibits robustness to variations in α.

6.3 Analysis of violation loss

Additionally, we conducted training with various versions of $\mathcal{L}_{viol}$. To ensure our objective remained unaffected, we tested three additional different loss functions. $\mathcal{L}_1$ is our original loss function in Eq. (5), and $\mathcal{L}_2$, $\mathcal{L}_3$, $\mathcal{L}_4$ is defined as follows:

$$\mathcal{L}_2 = \max(\text{viol}(\mathbf{x}, \mathbf{y}), 0)$$

$$\mathcal{L}_3 = \max(\text{viol}(\mathbf{x}, \mathbf{y}), 0) - \beta \cdot \min(\text{viol}(\mathbf{x}, \mathbf{y}), 0)$$

$$\mathcal{L}_4 = \begin{cases} \text{viol}(\mathbf{x}, \mathbf{y}) & \text{if viol}(\mathbf{x}, \mathbf{y}) \geq 0 \\ \exp(\text{viol}(\mathbf{x}, \mathbf{y})) - 1 & \text{otherwise} \end{cases}$$

While $\mathcal{L}_2$ most directly resembles the idea of our original objective of Eq. (4), we opted for the square of $\mathcal{L}_2$ to enhance training efficiency. On the other hand, $\mathcal{L}_3$ and $\mathcal{L}_4$ are the variants that keep minimizing $\text{viol}(\mathbf{x}, \mathbf{y})$ even when it is negative, but with a slower rate. The test results, shown in Figure 8, show no substantial differences, also confirming that our method is robust to variations in the loss function.

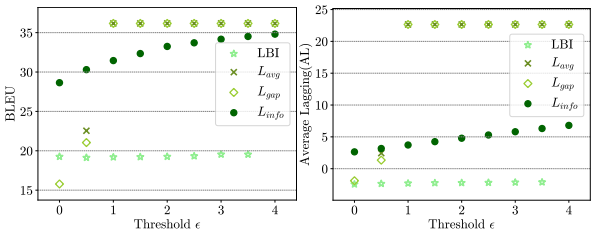


Figure 9: Performance comparison with different strategies to avoid degenerate solution on De → En.

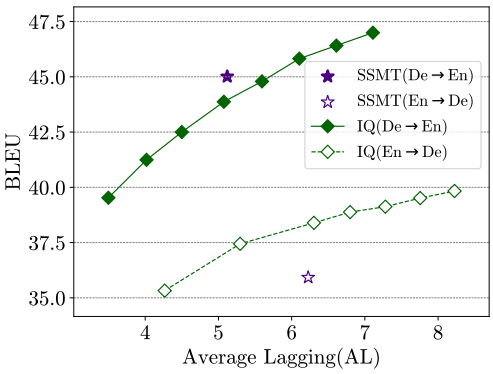


Figure 10: Train performances of SSMT and IQ, showing the improved generalization ability of IQ.

6.4 Importance of $\mathcal{L}_{info}$

In Sec 4.2, we introduced an auxiliary loss $\mathcal{L}_{info}$ to ensure that IQ does not converge to the degenerate solution where all the information of tokens is zero. However, such an auxiliary loss can be designed in many different ways, and we conducted additional experiments to see the effectiveness of proposed $\mathcal{L}_{info}$. We compare the following three different strategies:

Lower bounding information In this strategy, we applied $1+$*softplus* activation only to $\mathbf{IQ}^{\mathbf{src}}$ network to lower bound the source token's information to 1. While it has another degenerate solution where all tokens' information is 1, it is much harder to converge to it. We denote this strategy as LBI in Figure 9.

Equating length independently Similar to (Zhang et al., 2022), we used the following auxiliary loss in this strategy:

$$\mathcal{L}_{avg} = \|\mathbf{Info}^{\mathbf{src}}(\mathbf{x}) - \eta\|^2 + \|\mathbf{Info}^{\mathbf{trg}}(\mathbf{y}) - \eta\|^2$$

where $\eta = \frac{n+m}{2}$. With $\mathcal{L}_{avg}$, we are trying to equate the amount of information of source sentences and the amount of information of target sentences to the half number of all tokens. Unlike $\mathcal{L}_{info}$, this loss strongly suppresses the expressivity of the framework as we increase α since we make decisions based on the difference between $\mathbf{Info}^{\mathbf{src}}$ and $\mathbf{Info}^{\mathbf{trg}}$.

Encouraging margins In this strategy, to not suppress the expressivity of the framework and avoid degenerate solution at the same time, we aim to encourage gaps between the amounts of information of source/target sentences, making the decisions clearer. To this end, we define $\mathcal{L}_{gap}$ in such a way as to make the difference between $\mathbf{Info}^{\mathbf{src}}$ and $\mathbf{Info}^{\mathbf{trg}}$ larger than a constant value. We denoted this new definition as $\mathcal{L}_{gap}$, which can be formulated as follows:

$$\mathcal{L}_{gap} = \max(c - \mathbf{Info}^{\mathbf{gap}}(\mathbf{x}, \mathbf{y}), 0),$$

where c is the constant that defines the desired gap, and

$$\mathbf{Info}^{\mathbf{gap}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{Info}^{\mathbf{trg}}(\mathbf{y}) - \mathbf{Info}^{\mathbf{src}}(\mathbf{x})\|^2.$$

The test results are shown in Figure 9. While different strategies are showing comparable performance to each other (considering both BLEU and AL), the proposed alternative strategies are mostly either having very low-quality translation with small AL or fully offline translation with high AL. It demonstrates that using $\mathcal{L}_{info}$ not only avoids degenerate solution but also stabilizes the scale of differences between $\mathbf{Info}^{\mathbf{src}}$ and $\mathbf{Info}^{\mathbf{trg}}$ unlike other methods, such that the quality-latency trade-off is controllable with ϵ.

6.5 Generalization ability

Lastly, we demonstrate the improvement of the generalization ability of our framework. We utilize a sample of 6K instances from the training set and additionally compare the performance of SSMT and IQ. The results presented in Figure 10 indicate that SSMT, which trains a READ/WRITE policy directly from oracle action sequences, performs on par with IQ on the training set, unlike the test set performances. As we observed in the main experiments, SSMT shows relatively lower test performances compared to IQ, implying that IQ less over-fits and possesses better generalization ability due to the clever design of the framework.

7 Conclusion

In this paper, we introduced a novel framework of training and inferencing with Information Quantifier (IQ) for Simultaneous Translation (ST) by

using oracle action sequences. We demonstrated that IQ exhibits high performance despite its simplicity and flexibility, being able to adapt to various latency regimes with a single model.

Limitations

We employed the strategy of accepting WRITE actions when the online decoding token is the same as the offline decoding token as suggested by SSMT to generate oracle action sequences. While we demonstrated IQ framework is more robust to different action sequence generations compared to SSMT, degradation of performance is inevitable when the given action sequences are far from optimal. Since obtaining optimal action sequences is expensive in many cases, the proposed framework will be hard to apply when the oracle action sequence generation heuristics suggested by SSMT do not perform well.

Acknowledgements

This work was supported by SAMSUNG Research, Samsung Electronics Co., Ltd. This work was also partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00311, Development of Goal-Oriented Reinforcement Learning Techniques for Contact-Rich Robotic Manipulation of Everyday Objects) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1F1A1074880)

References

Ashkan Alinejad, Hassan S. Shavarani, and Anoop Sarkar. 2021. Translation-based supervision for policy generation in simultaneous neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1734–1744, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2015. The IWSLT 2015 evaluation campaign. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–14, Da Nang, Vietnam.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2013. Report on the 10th IWSLT evaluation campaign. In *Proceedings of the 10th International Workshop on Spoken Language Translation: Evaluation Campaign*, Heidelberg, Germany.

Chung-Cheng Chiu and Colin Raffel. 2018. Monotonic chunkwise attention. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Qian Dong, Yaoming Zhu, Mingxuan Wang, and Lei Li. 2022. Learning when to translate for streaming speech. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–694, Dublin, Ireland. Association for Computational Linguistics.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient wait-k models for simultaneous machine translation. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 1461–1465. ISCA.

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.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 92–105, New Orleans - Louisiana. Association for Computational Linguistics.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-latency sequence-to-sequence speech recognition and translation by partial hypothesis selection. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 3620–3624. ISCA.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Thai-Son Nguyen, Sebastian Stüker, and Alex Waibel. 2021. Super-human performance in online low-latency recognition of conversational speech. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 1762–1766. ISCA.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Does simultaneous speech translation need simultaneous models? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 141–153, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sara Papi, Matteo Negri, and Marco Turchi. 2023. Attention as a guide for simultaneous speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356, Toronto, Canada. Association for Computational Linguistics.

Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. 2022. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 277–285, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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. 2021. Universal simultaneous machine translation with mixture-of-experts wait-k policy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7306–7317, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022a. Gaussian multi-head attention for simultaneous machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3019–3030, Dublin, Ireland. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2022b. 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*.

Shaolei Zhang, Shoutao Guo, and Yang Feng. 2022. Wait-info policy: Balancing source and target at information level for simultaneous machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2249–2263, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2847–2853, Online. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, Hong Kong, China. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822, Florence, Italy. Association for Computational Linguistics.