The University of Edinburgh's Submission to the IWSLT21 Simultaneous Translation Task

Sukanta Sen, Ulrich Germann and Barry Haddow
University of Edinburgh
{ssen, ugermann, bhaddow}@inf.ed.ac.uk

Abstract

We describe our submission to the IWSLT 2021 shared task[1] on simultaneous text-to-text English-German translation. Our system is based on the re-translation approach where the agent re-translates the whole source prefix each time it receives a new source token. This approach has the advantage of being able to use a standard neural machine translation (NMT) inference engine with beam search, however, there is a risk that incompatibility between successive re-translations will degrade the output. To improve the quality of the translations, we experiment with various approaches: we use a fixed size wait at the beginning of the sentence, we use a language model score to detect translatable units, and we apply dynamic masking to determine when the translation is unstable. We find that a combination of dynamic masking and language model score obtains the best latency-quality trade-off.

1 Introduction

In spoken language translation (SLT), there is often a need to produce translations *simultaneously*, without waiting for the speaker to finish. For example, we may be targeting live events such as conferences or meetings where excessive latency will disrupt the user experience. In order to achieve low latency SLT, however, translation systems must be able to cope well with incomplete utterances, and we find that we need to trade off latency for translation quality. In research on simultaneous SLT, we would like to understand how to produce the best possible trade-off between these two measures. In the IWSLT 2021 shared task on simultaneous translation, the aim was to build and evaluate simultaneous SLT systems at three different latency regimes (low, medium and high), as measured using the Average Lagging (AL; Ma et al. (2019)).

There are two main approaches to simultaneous translation: streaming (Cho and Esipova, 2016; Ma et al., 2019) where the system appends the output to a growing hypothesis as new inputs are available, and re-translation (Niehues et al., 2016, 2018; Arivazhagan et al., 2020a,b), where, as the name suggests, the system re-translates the whole prefix on every update to a completely new output. Re-translation approach has the advantage that we can use an unmodified, general purpose, optimised MT engine with beam-search, but we have to address the problem of *flicker*. That is to say, the translation of a prefix may be changed by the translation of an extended prefix. Recent work by Arivazhagan et al. (2020a) has shown that, if measures are taken to mitigate flicker, then re-translation produces results comparable to streaming approach. Since the shared task does not permit any revision of a committed hypothesis (i.e. flicker is not allowed) we focus on adapting the re-translation approach for our submission without introducing any flicker into a growing hypothesis.

2 Overview of Our Submission

We participated in the English→German text-to-text simultaneous task. Since we re-translate the incomplete input (know as a prefix) each time it is updated, our system will try to modify the translations produced from earlier prefixes. But as the task is evaluated using SimulEval (Ma et al., 2020) which does not permit the modification of committed output (also known as flickering), we use a simple approach to generate incremental output at each re-translation step.

Concretely, we apply a method inspired by the wait-k streaming approach (Ma et al., 2019) in our re-translation system in the following manner. In the task, a simultaneous SLT system is implemented as an agent which must choose between

[1] https://iwslt.org/2021/

Proceedings of the 18th International Conference on Spoken Language Translation, pages 46–51
Bangkok, Thailand (Online), August 5–6, 2021. ©2021 Association for Computational Linguistics

READ (read more input) and WRITE (append to the current translation hypothesis) operations. Our overall approach is shown in Algorithm 1. The agent first performs k consecutive READ operations and then alternatively READs and WRITEs until the full input sentence is read. Once the input is consumed, the agent keeps performing WRITE operations until it reaches the end of the translated sentence. The WRITE operation involves re-translating the prefix S and finding the next output word w from output prefix T. If the output prefix T has a length longer than the committed hypothesis H, it picks the $(i+1)$th word of T, else sends READ signal to the agent, i being the length of the current hypothesis.

Algorithm 1 Our Re-translation Approach

```
Require: NMT system $\phi$, $k$
 1: Initialize: $S \leftarrow \{\}, H \leftarrow \{\}, w \leftarrow \varepsilon$
 2: while $w$ is not ⟨eos⟩ do
 3:   if $|S| - |H| < k$ and not finished reading then
 4:     READ next input $s$
 5:     $S \leftarrow S \cup \{s\}$
 6:   else
 7:     $T \leftarrow \phi(S)$
 8:     if $|T| > |H|$ then
 9:       $w \leftarrow T[|H| + 1]$
10:     else
11:       $w \leftarrow \varepsilon$
12:     end if
13:     if $w$ is not $\varepsilon$ or finished reading then
14:       $H \leftarrow H \cup \{w\}$
15:       WRITE $w$
16:     end if
17:   end if
18: end while
```

However, there is a potential problem with this approach. In each WRITE step, the output word w is selected from the $(|H|+1)$th position of output prefix T. Thus if any correction is made by a re-translation in the initial $|H|$ words, the WRITE operation won't be able to recover the mistake. In other words, our approach is able to suppress the flicker caused by re-translation, but could end up gluing together incompatible fragments of the hypothesis. This problem can be worse when the output prefix T flickers too much. To improve translation quality, we employ two approaches which aim at detecting meaningful units (MU) and allowing extra READs when inside an MU. An MU is a chunk of words that has a definite translation and can be translated independently without having to wait for more input words (Zhang et al., 2020).

Our first method of detecting MUs relies on the language model (LM) score. The agent keeps track of the language model (LM) score of the previous token and compares it with the score of the current token. If the LM score is higher than the previous token, it keeps reading more tokens and does a re-translation only when this condition is not met. Here the LM score is the log probability of the current token given the context. Though LM score doesn't guarantee to find meaningful unit every time but this simple approach shows it is better than the baseline approach in terms of BLEU score.

Our second method of stabilising the re-translation approach is based on the idea of dynamic masking (Yao and Haddow, 2020). The dynamic mask approach finds the stable part of the target prefix by comparing the translation of the current prefix, with the translation of an extension of the current prefix. The longest common prefix (LCP) of the two translations is taken as the stable part. Figure 1 shows how dynamic masking works in general. Yao and Haddow (2020) showed that using dynamic mask could give a better flicker-latency trade-off than using a fixed mask, without affecting the translation quality of full sentences.

For our IWSLT submission, we generate the extended prefixes for dynamic mask simply by appending *UNK* (i.e the unknown word symbol) to the prefix. In Figure 2, we show an example of how dynamic mask stabilises the translation, by masking the least stable part of the MT output. This translation-with-dynamic-mask provides a drop-in replacement for the MT system $\phi()$ in line 7 of Algorithm 1, except when the agent has read the full input sentence, when we do not need to apply any mask.

3 Experimental Details

We use only the officially allowed IWSLT 2021 data sets. The training data include high quality English-German parallel data from WMT 2020 (Barrault et al., 2020), English-German data from MuST-C.v2 (Di Gangi et al., 2019), the TED corpus (Cettolo et al., 2012) and OpenSubtitle (Lison and Tiedemann, 2016). For development, we use the concatenation of IWSLT test sets from 2014 and 2015. We test on IWSLT 2018 test set and tst-

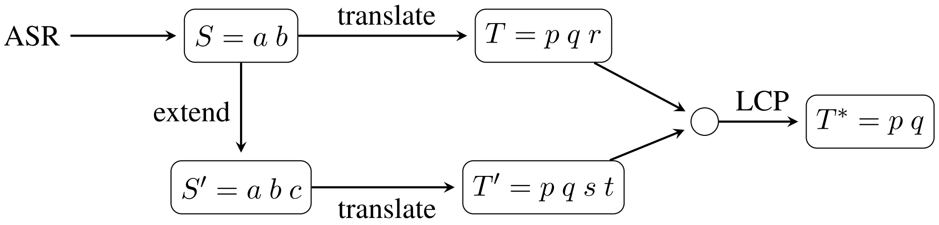


Figure 1: Dynamic Masking. The string $a\ b$ is provided as input to the agent (in a full SLT system it would come from ASR). The MT system then produces translations of the string and its extension, compares them, and outputs the longest common prefix (LCP)

	Source	Translation	MT Output
prefix	Back in New York,	Zurück in New York,	
extension	Back in New York, UNK	Damals in New York, in	
prefix	Back in New York, I	Damals in New York have ich	
extension	Back in New York, I UNK	Damals in New York war I	Damals in New York

Figure 2: An example of dynamic mask applied during translation. For the first prefix, the translation of the prefix and its extension disagree, so no output is produced (i.e. all output is masked). For the second prefix, the translation is more stable.

COMMON from MuST-C.v2. As the there is a significant overlap between MuST-C.v2 and tst-20{14,15,18}, we remove the overlaps from the MuST-C.v2 training data before training.

For preprocessing we rely only on SentencePiece tokenization (Kudo and Richardson, 2018); no other preprocessing tools are applied. We use a shared vocabulary size of 32k. Standard NMT models perform well when translation is done on a full sentence but as our approach is based on retranslation, we use training data that is a 1:1 mix of full sentences and prefix pairs (Niehues et al., 2018; Arivazhagan et al., 2020a). This ensures that our model can translate both full sentences and prefixes. To create prefix pairs, we first randomly choose a position in the source sentence and then take the proportionate length of the target sentence. Along with that we also add modified prefix pairs in which the source side has a shorter target prefix appended with the source prefix. The purpose of these modified prefix pairs was to investigate an alternative type of stabilisation, where the previous target prefix is fed into the translation of the current source prefix, but in early testing this method did not work well, so we did not pursue it further. The validation data is also pre-processed similarly to the training set. Note that this preprocessed validation set is used at training for early stopping and not for reporting the validation scores in the Table 2.

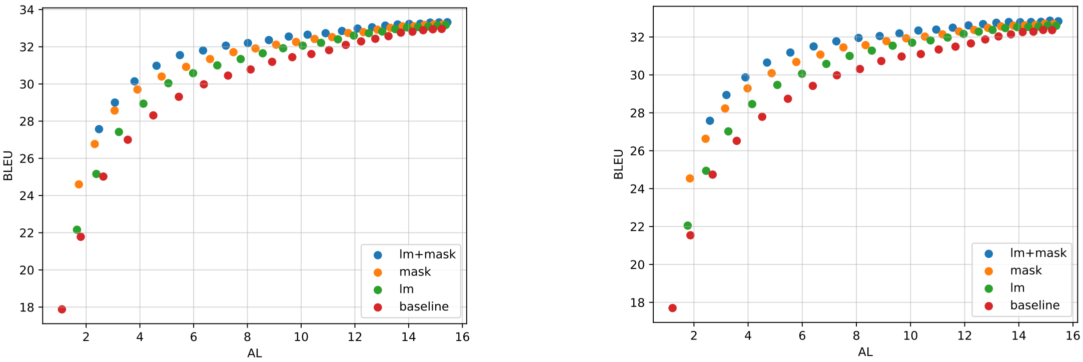
For training, we use the Marian toolkit (Junczys-Dowmunt et al., 2018) with the 'base' transformer architecture (Vaswani et al., 2017). First, we train a model using the aforementioned pre-processed training data and then fine-tune the model using MuST-C.v2 training data which is more of a domain specific data for simultaneous translation task. To train the language model for stabilisation, we use KenLM (Heafield, 2011) to train a 6-gram language model on the source-side training data. We have shown the number of sentences in each corpus in Table 1.

Corpus	Sentence pairs
Europarl	1.79 M
Rapid	1.45 M
News Commentary	0.35 M
OpenSubtitle	22.51 M
TED corpus	206 K
MuST-C.v2	248 K

Table 1: Corpora used in training the systems

4 Result and Analysis

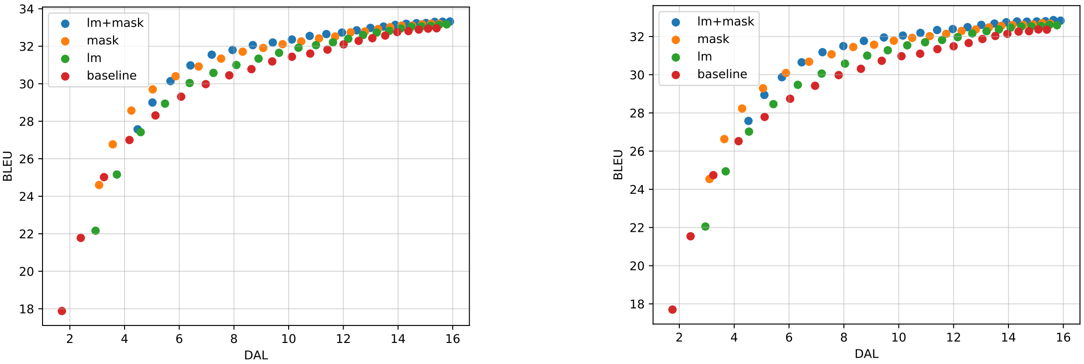
We evaluate the model's performance on the full sentence translation before doing actual simultaneous translation. For this evaluation we use SacreBLEU (Post, 2018) on the MuST-C.v2 and TED 2018 test sets. The results on full sentence is shown in the Table 2. We see there is a significant improve-



(a) Beam size = 12, Normalization = 1.0

(b) Beam size = 12, Normalization = 0.6

Figure 3: BLEU vs AL plots for English-German with different beam sizes and length normalization.



(a) Beam size = 12, Normalization = 1.0

(b) Beam size = 12, Normalization = 0.6

Figure 4: BLEU vs DAL plots for English-German with different beam sizes and length normalization.

ment after fine-tuning. For full sentence (or prefix in case of re-translation) translation we set beam size 12 and length normalization 1.0 in Marian.

	Validation	Test	
	TED 2014,15	TED 2018	MuST-C.v2
Baseline	30.8	27.5	32.7
Fine-tuned	31.9	29.4	33.6

Decoder settings: Beam size = 12; Normalization = 1.0

Table 2: BLEU scores on full sentence translation, computed with SacreBLEU.[a]

[a] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.1

For evaluating the simultaneous translation, we use SimulEval (Ma et al., 2020) which calculates SacreBLEU for quality and Average Lagging (AL) (Ma et al., 2019), differential AL (DAL) (Cherry and Foster, 2019), and average proportion (AP) (Cho and Esipova, 2016) for latency. The official evaluation uses a blind test set, however, for submission purpose, we evaluate it on the MuST.v2 test set (tst-COMMON) set. We have following settings for re-translation:

Type	k	AL	BLEU	Approach
Full Sentence	-	-	33.60	-
High	20	14.73	33.09	lm
High	21	14.94	33.2	mask
High	20	14.8	**33.3**	lm+mask
Medium	6	5.98	30.58	lm
Medium	6	5.72	30.92	mask
Medium	5	5.49	**31.55**	lm+mask
Low	2	2.38	25.16	lm
Low	2	2.32	26.77	mask
Low	1	2.48	**27.57**	lm+mask

Table 3: AL vs BLEU scores for three regimes (Low, Medium, High) on MuST-C.v2 test set using beam size 12 and normalization 1.0. Best scores are in bold.

- *baseline*: The agent waits for initial k tokens and then alternates between READ and WRITE (using re-translation). This is similar to the wait-k approach by Ma et al. (2019).

- *lm*: After the initial k tokens, the agent uses the language model to determine the "meaningful unit" boundaries, and only WRITEs when at a boundary.

- *mask*: This is similar to the baseline, except that the agent applies dynamic masking to produce a more stable translation.
- *lm*+*mask*: Combination of *lm* and *mask*. Thus in this approach, the agent first uses the *lm* score to decide whether to translate, and then uses dynamic mask to obtain a more stable translation.

The official evaluation has three regimes of latency: low (AL$\leq$ 3), medium (AL$\leq$ 6) and high (AL$\leq$ 15). In Table 3, we show the AL and BLEU scores for the three regimes with different approaches. We find that LM score and Dynamic masking combined achieve the best AL-BLEU trade-off.

To gain a fuller comparison of approaches, we calculate AL vs. BLEU and DAL vs. BLEU for a range of k values, and different stabilisation approaches and plot them as shown in Figures 3 and 4. Whilst for any given k, the *lm*+*mask* approach has higher AL (because it adds WAIT operations), we can see from the trajectory of the plot in Figure 3 that the *lm*+*mask* approach has the best AL-BLEU trade-off. While training the models, we set the length normalization to 0.6 which is used for scoring the development set for the purpose of early-stopping. However, we find that a normalization 1.0 performs slightly better than normalization 0.6 when doing re-translation. We show the plots for both normalization values in figures 3 and 4.

When the AL is 15, for many sentences it is a full sentence translation and thus all the approaches have similar BLEU scores. We also notice many sentences have negative AL scores. As the corpus AL scores is the average of the sentence level AL scores, negative scores can reduce the actual AL score. To address this shortcoming of AL, Cherry and Foster (2019), propose *Differentiable Average Lagging* (DAL) as an alternative. In Figure 4, we show the DAL vs BLEU scores. In Figure 4, we also observe that the proposed LM and masking improve the baseline by a significant margin in DAL-BLEU trade-off.

5 Conclusion

In this paper, we describe our submission to the IWSLT 2021 shared task on simultaneous text-to-text German-English translation. We work with a re-translation approach, enabling use to use an unmodified MT inference engine, together with an adaptation of wait k to trade off quality and latency. Additionally we proposed two techniques (dynamic masking and LM score) to improve translation quality by reducing the potential for flicker. We find that the combination of the proposed approaches achieves the best AL-BLEU trade-off.

Acknowledgments

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (`www.dirac.ac.uk`).

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreements No 825460 (Elitr) and No 825627 (European Language Grid).

References

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, and George Foster. 2020a. Re-translation versus streaming for simultaneous translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 220–227, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Isabelle Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2020b. Re-translation strategies for long form, simultaneous, spoken language translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7919–7923. IEEE.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Conference of european association for machine translation*, pages 261–268.

Colin Cherry and George Foster. 2019. Thinking slow about latency evaluation for simultaneous machine translation. *arXiv preprint arXiv:1906.00048*.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012*.

Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Must-c: a multilingual speech translation corpus. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2012–2017. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

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. 2020. Simuleval: An evaluation toolkit for simultaneous translation. In *Proceedings of the EMNLP*.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2016. Dynamic transcription for low-latency speech translation. In *Interspeech*, pages 2513–2517.

Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel. 2018. Low-latency neural speech translation. *arXiv preprint arXiv:1808.00491*.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Yuekun Yao and Barry Haddow. 2020. Dynamic masking for improved stability in online spoken language translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 123–136, Virtual. Association for Machine Translation in the Americas.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289, Online. Association for Computational Linguistics.