Modeling Context With Linear Attention for Scalable Document-Level Translation

Zhaofeng Wu[♋] **Hao Peng**[♌] **Nikolaos Pappas**[♊] **Noah A. Smith**[♌♍]
[♋]MIT [♌]Allen Institute for Artificial Intelligence [♊]AWS AI
[♍]Paul G. Allen School of Computer Science & Engineering, University of Washington
zfw@csail.mit.edu haop@allenai.org
nppappa@amazon.com nasmith@cs.washington.edu

Abstract

Document-level machine translation leverages inter-sentence dependencies to produce more coherent and consistent translations. However, these models, predominantly based on transformers, are difficult to scale to long documents as their attention layers have quadratic complexity in the sequence length. Recent efforts on efficient attention improve scalability, but their effect on document translation remains unexplored. In this work, we investigate the efficacy of a recent linear attention model by Peng et al. (2021) on document translation and augment it with a sentential gate to promote a recency inductive bias. We evaluate the model on IWSLT 2015 and OpenSubtitles 2018 against the transformer, demonstrating substantially increased decoding speed on long sequences with similar or better BLEU scores. We show that sentential gating further improves translation quality on IWSLT.[1]

1 Introduction

Sentence-level neural machine translation has seen significant recent progress (Bahdanau et al., 2015; Vaswani et al., 2017). A move to document-level translation opens the possibility of using inter-sentential context at the scale of paragraphs, documents, or even whole books (Lopes et al., 2020; Ma et al., 2021b; Maruf et al., 2021). This opens up new research avenues to improve translation and its evaluation for more consistent anaphora resolution and discourse coherence (Bawden et al., 2018; Müller et al., 2018; Voita et al., 2019).

Transformers have enabled state-of-the-art results for sentence-level translation (Vaswani et al., 2017; Chen et al., 2018; Wang et al., 2019) and this success has made them the default architecture for document translation. However, they do not scale well in the sequence length due to the quadratic complexity of attention and hence are computationally prohibitive to apply to long text. Alternative architectures exist, but most still have quadratic complexity (Zhang et al., 2018; Voita et al., 2019) and/or have extra modules that further add to the inference cost (Tu et al., 2018; Zhang et al., 2018; Miculicich et al., 2018; Donato et al., 2021).

By reducing asymptotic complexity, recent work on efficient attention may pave the way for long text generation. However, these methods' suitability for document translation requires further investigation: some do not focus on decoding speed (Guo et al., 2019; Child et al., 2019; Kitaev et al., 2020; Wang et al., 2020, *i.a.*), while others' speedup and quality impact for document translation remains unknown (Kasai et al., 2021; Schlag et al., 2021; Ma et al., 2021a; Choromanski et al., 2021, *i.a.*). In this work, we consider random feature attention (RFA; Peng et al., 2021), a representative first model with established accuracy and efficiency in sentence-level translation. With few additional parameters, it approximates softmax attention in linear time and space with recurrent computations. We explore its effectiveness for document translation and find substantial decoding speedup over a transformer with similar or improved BLEU. We also equip RFA with a sentential gate, injecting a recency inductive bias tailored to representing document context.

Our main contributions are: (i) we study the efficacy of RFA for document translation; (ii) we validate that RFA is competitive with a transformer but substantially faster on long documents; (iii) we augment RFA with a sentential gate designed to promote a recency bias, which brings a 0.5 BLEU improvement on IWSLT (Cettolo et al., 2015). To encourage future research, we release our code.[1]

2 Background

Standard machine translation independently translates each sentence. Document translation leverages additional source and target context to produce

[1] https://github.com/ZhaofengWu/rfa-doc-mt

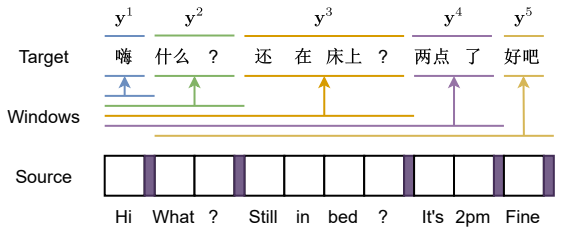

Figure 1: The concatenation model for document translation with a sliding window of length $L = 4$. Every window is translated in its entirety, but only the last translated sentence is used for evaluation. The purple bars denote the sentence separator token.

more coherent translation, improving lexical choice and ambiguity resolution (Voita et al., 2019).

The Concatenation Model. Recent studies found that the concatenation model that directly translates the source document to the target document (or a multi-sentence window) with a single encoder-decoder model performs well (Tiedemann and Scherrer, 2017; Ma et al., 2021b), especially on large datasets (Junczys-Dowmunt, 2019). Figure 1 illustrates this model combined with sliding window decoding, which we adopt in this work.

Scalability of Attention. In machine translation, transformers have three types of attention: encoder self-attention, cross attention, and causal attention. In each, every query $\mathbf{q}_t$ is dotted with all keys $\{\mathbf{k}_i\}$ to obtain the attention weights, with which a weighted average of the values $\{\mathbf{v}_i\}$ is calculated:

$$\text{attn}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right) = \sum_{i=1}^{N} \frac{\exp\left(\mathbf{q}_t \cdot \mathbf{k}_i\right)}{\sum_{j=1}^{N} \exp\left(\mathbf{q}_t \cdot \mathbf{k}_j\right)} \mathbf{v}_i^\top$$

where N is the sequence length. This pairwise interaction incurs quadratic overhead in N, which is inefficient for the long text sequences in the concatenation model. This particularly impacts cross and causal attention at decoding time, which cannot be parallelized (Kasai et al., 2021).[2]

3 Scalable Document-Level Translation

We test RFA as a linear time and space model to improve the efficiency of document translation. We also augment it with a sentential gate to circumvent capacity constraints that come with a long context by injecting a recency inductive bias.

[2]As an example, IWSLT has a ≈ 80 sequence length with a window size of 4 (Table 2, appendix).

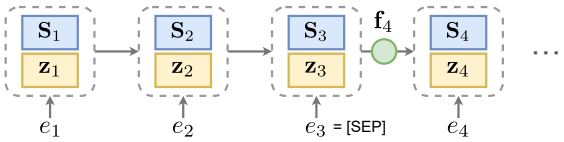

Figure 2: Our sentential gating mechanism. e_1 and e_4 are at the beginnings of two sentences.

3.1 Random Feature Attention

RFA approximates the softmax attention $\text{attn}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right)$ in linear time and space:

$$\text{RFA}\left(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}\right) = \frac{\boldsymbol{\phi}\left(\mathbf{q}_t\right) \cdot \mathbf{S}_t}{\boldsymbol{\phi}\left(\mathbf{q}_t\right) \cdot \mathbf{z}_t}.$$

The randomized nonlinear transformation $\boldsymbol{\phi}(\cdot)$ is constructed so that $\exp(\mathbf{x} \cdot \mathbf{y}) \approx \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{y})$ (Rahimi and Recht, 2008). $\mathbf{S}$, $\mathbf{z}$ summarize keys and values. We use RFA in cross and causal attention, which are the most impactful for speed and memory, so $\mathbf{q}_t$ is always from the target sequence. In cross attention, $\mathbf{S}$ and $\mathbf{z}$ represent the source sequence and are constant for all query positions t: $\mathbf{S}_t = \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{k}_i)\mathbf{v}_i^\top$ and $\mathbf{z}_t = \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{k}_i)$. In causal attention, they represent the target prefix $i \leq t$: $\mathbf{S}_t = \sum_{i=1}^{t} \boldsymbol{\phi}(\mathbf{k}_i)\mathbf{v}_i^\top = \mathbf{S}_{t-1} + \boldsymbol{\phi}\left(\mathbf{k}_t\right)\mathbf{v}_t^\top$ and $\mathbf{z}_t = \sum_{i=1}^{t} \boldsymbol{\phi}(\mathbf{k}_i) = \mathbf{z}_{t-1} + \boldsymbol{\phi}\left(\mathbf{k}_t\right)$. These recurrent computations are analogous to an RNN with $\mathbf{S}_t$ and $\mathbf{z}_t$ as hidden states at step t and enable constant computation per timestep. RFA serves as a drop-in replacement for attention in transformers. The encoder and other modules, e.g., feed-forward layers, remain the same. We refer the reader to Peng et al. (2021) for more details on RFA.

3.2 Sentential Gating

Schlag et al. (2021) noted, under the lens of fast weight programmers (Schmidhuber, 1991, 1992, 1993), that accumulating memory in a purely additive manner, like $\mathbf{S}$ and $\mathbf{z}$ above, will reach a capacity limitation with sequences longer than the size of $\boldsymbol{\phi}$. This is particularly an issue in document-level translation due to the long sequences.

Nevertheless, document translation admits a natural solution to this constraint: inspired by gated RNNs (Cho et al., 2014, *i.a.*), we augment RFA with a sentence-level gate to enable dynamic control of contextual information from the current and previous sentences, and to allow the model to selectively forget about the history to circumvent the capacity constraint. This is illustrated in Figure 2. For the tth word with representation $\mathbf{e}_t$, we compute a

forget gate using the sentence separator token:

$$f_t = \begin{cases} \sigma(\mathbf{w}_f \cdot \mathbf{e}_{t-1} + b_f) & \text{if } x_t \text{ starts a sentence} \\ 1 & \text{otherwise} \end{cases}$$

$$\mathbf{S}_t = f_t\, \mathbf{S}_{t-1} + \boldsymbol{\phi}\,(\mathbf{k}_t)\, \mathbf{v}_t^\top$$

$$\mathbf{z}_t = f_t\, \mathbf{z}_{t-1} + \boldsymbol{\phi}\,(\mathbf{k}_t)$$

σ is the sigmoid function; $\mathbf{w}_f$ and b_f are learned parameters. The context is decayed when a new sentence starts, and the decay coefficient is reused for all tokens in the same sentence. Specifically, each sentence j assigns a weight $0 < \prod_{i=\text{START}(j')+1}^{\text{START}(j)} f_i < 1$ when attending to a previous sentence j', where $\text{START}(\cdot)$ is the first token in a sentence. This introduces an inductive bias that, intuitively, previous sentences are less important in translation, and their representations are decayed.

Relation to Prior Work. While gating is common in RNNs, it is less clear how it applies to transformers. Miculicich et al. (2018) gated at the sentence level, though hierarchically, while we gate recurrently. Ours also contrasts with the per-token gating of Peng et al. (2021) which they found ineffective for machine translation. These two works also take a weighted average of the previous and current sentences while we only decay the former. We show our variant performs better in §5. Schlag et al. (2021) used a gate that explicitly models memory removal, but also at the token level.

4 Experimental Setup

Datasets and Evaluation. We experiment with the IWSLT 2015 Chinese-to-English (zh-en) dataset (Cettolo et al., 2015) with multilingual TED talk captions and the OpenSubtitles2018 English-to-Russian (en-ru) dataset (Lison et al., 2018) with movie and TV subtitles. We measure document-level BLEU (Papineni et al., 2002) with SacreBLEU (Post, 2018).[3] To quantify discourse consistency, we also use the test sets by Voita et al. (2019) based on OpenSubtitles. We introduce these datasets and their statistics in more detail in §A.1.

Data Processing. We process each document with a stride-one sliding window of L sentences to obtain our training set. Following Voita et al. (2019) and Ma et al. (2021b), we experiment with $L = 1$, the sentence-level baseline, and $L = 4$.

[3]We use fairseq's default setting which has hash `case.mixed+numrefs.?+smooth.exp+tok.none+version.1.5.0` with standalone 13a-tokenization.

	IWSLT		Subtitles	
Window Size L	**1**	**4**	**1**	**4**
Transformer	31.7	30.4	32.6	33.1
RFA	31.0	**30.7**	**32.9**	**33.2**
RFA-sgate-avg	—	**30.8**	—	33.0
RFA-sgate	—	**31.2**	—	**33.2**

Table 1: BLEU on IWSLT and OpenSubtitles test sets. Bold scores outperform the transformer.

During inference, we use the last translated sentence in each window for evaluation. For a more granular analysis, we consider $L \in \{1, 2, 3, 4\}$ for consistency experiments. More details are in §A.1.

Model Settings. We compare RFA and transformer with the concatenation model. For RFA, we experiment with no gating (**RFA**) and sentential gating (**RFA-sgate**). To compare our decaying gate choice with prior work (§3.2), we run a sentential-gated RFA that takes a weighted average of previous and current text (**RFA-sgate-avg**). We mostly default to fairseq hyperparameters (Ott et al., 2019), most suitable for the $L = 1$ transformer (§A.2).

5 Results

BLEU Scores. Table 1 shows BLEU scores on IWSLT and OpenSubtitles. The sentence-level transformer has the highest IWSLT BLEU, possibly due to defaulting to fairseq hyperparameters optimized for this setting. With $L = 4$, RFA performs slightly better than the transformer, showing suitability for long-text translation. Gated RFA further brings a 0.5 BLEU improvement, confirming its utility, but gating has no effect on OpenSubtitles. We hypothesize that with only ≈10 tokens per sentence, half of the average length of IWSLT (Table 2, appendix), the capacity constraint (Schlag et al., 2021) is less severe and thus gating would be less useful. Our gate also outperforms the averaging variant in Miculicich et al. (2018) and Peng et al. (2021), validating its effect on document translation. Aligning with prior findings (Voita et al., 2019; Ma et al., 2021b), longer contexts do not clearly lead to better BLEU, though it improves consistency metrics, to which we turn next.

Discourse Consistency Scores. Figure 3 plots the consistency scores in four phenomena for RFA, including our gated variants, and the transformer baseline from Voita et al. (2019) and Ma et al.

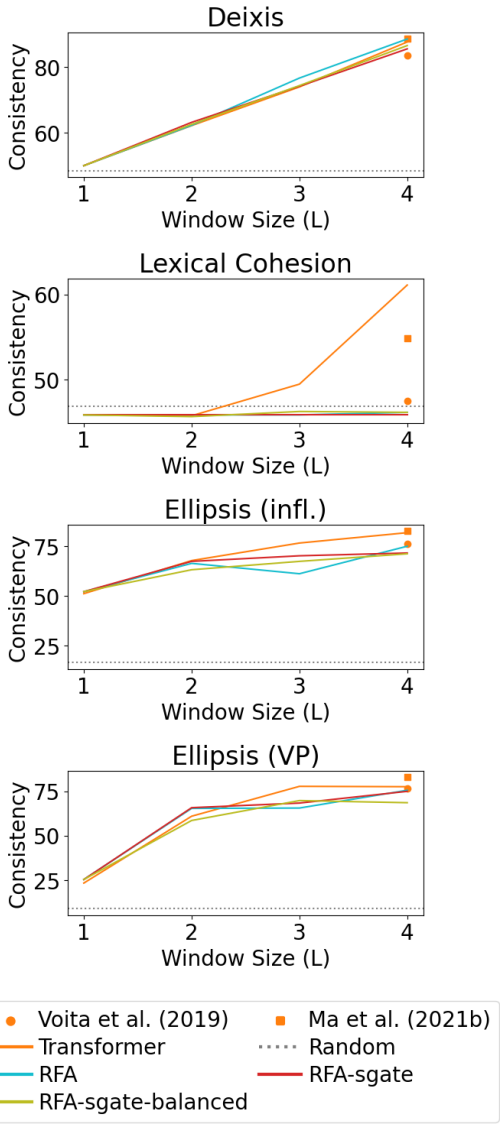

Figure 3: Model performance on the consistency test set, broken down into phenomena. Transformer and RFA are tested with window sizes from 1 to 4. We compare with the baselines in Voita et al. (2019) and Ma et al. (2021b) corresponding to our Transformer $L = 4$.

(2021b). We also re-implement this transformer model to control confounding factors in implementation details and to extrapolate to $L < 4$, which they did not thoroughly explore. We compare to a baseline that chooses its prediction randomly from candidate translations; see §A.1 for details.

Translating with longer contexts almost always yields better consistency, which is also a setting where RFA achieves better speedup, shown later. Gating does not have a clear benefit, aligning with OpenSubtitles' BLEU pattern. RFA slightly underperforms the transformer on ellipsis. We hypothesize that the direct query-key interaction in softmax attention is more suitable for precise long-distance

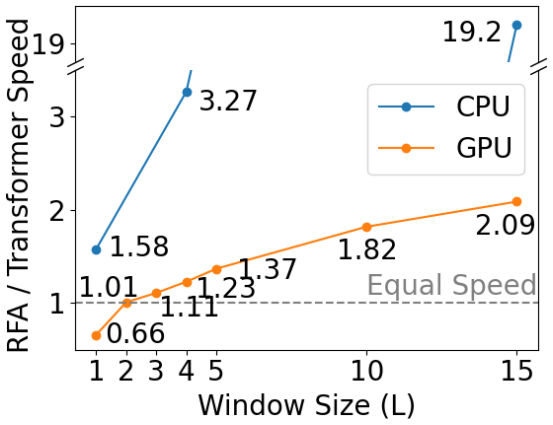

Figure 4: RFA's inference speedup over the transformer in the number of decoded tokens per second. Each sentence has ≈ 20 tokens (Table 2, appendix).

information extraction, sometimes required for consistency, than the RFA approximation. On lexical cohesion, transformer shows a large variance: with the same architecture and size, Ma et al. (2021b), Voita et al. (2019), and our implementation of $L = 4$ transformer achieve drastically different scores. Voita et al. (2019)'s implementation and RFA fail to outperform the random baseline on this phenomenon. Reliable evaluation of lexical cohesion, and the related task of word sense disambiguation, are known to be challenging in document translation: models tend to rely on dataset artifacts but not the context, and the attention of well-performing models poorly aligns with the ground-truth required for disambiguation (Kim et al., 2019; Emelin et al., 2020; Yin et al., 2021).

Speed. We confirmed the observation from prior work that longer context boosts translation consistency and sometimes BLEU. It would be exciting to examine this trend with $L > 4$, but to our knowledge, it has little existing evaluation data. We therefore measure decoding efficiency with a synthetic experiment by decoding for all L with the same trained model. We focus only on efficiency here, not quality. We measure the number of decoded tokens per second over the forward pass time on IWSLT's test set. We follow Ott et al. (2018) and cache $\mathbf{k}$ and $\mathbf{v}$ for our *baseline* which substantially increases its speed. More details are in §A.3.

Figure 4 shows RFA's speedup relative to the transformer. On GPU, without document context, RFA is slower, likely due to its random matrix overhead. Nevertheless, its speed over the transformer roughly linearly increases, agreeing with the theory, up to $2.09\times$ on our longest tested context $L = 15$.

RFA enables an even more substantial speedup on other device types. For sentence-level translation, RFA is in fact faster than the transformer by 1.58× on CPU, and, as Peng et al. (2021) reported, by 1.8–1.9× on TPU. At $L = 15$, its CPU speedup increases to 19.2×. Therefore, depending on the use case, such as when decoding on edge devices, RFA could be even more favorable. Furthermore, we used the same batch size between RFA and the transformer. With lower memory complexity, RFA accommodates a larger batch size and in practice achieve a more significant speedup. For example, at $L = 15$ on GPU, we found that RFA allows a 5× batch size and enables a more than 7× speedup.

RFA's superior speed makes it an attractive choice to leverage very long contexts. Nevertheless, we are merely extrapolating the utility of long context from our experiments. The extent to which it really helps needs to be verified by future curated test sets. We hope the demonstration of our model's ability to efficiently and effectively process document context could catalyze such efforts.

6 Conclusion

We explored the effectiveness of random feature attention on document translation. Our model substantially improves its speed over a transformer with similar or improved BLEU. Our sentential gate also proves effective, especially on long sequences. While our model may potentially be used to produce toxic or fake information, it also enables more efficient detectors toward such content.

Limitations

Limited by existing document translation datasets where "documents" are usually relatively short multi-sentence windows, we adopted a semi-synthetic setup for our speed benchmark experiments to examine RFA's effectiveness on long sequences. We believe our results should transfer to real data since decoding speed is mostly a function of sentence length, but this is not a guarantee. Additionally, while RFA would enjoy a better speedup on TPUs as reported in the original RFA paper, we did not have the necessary resources to run experiments on TPUs, so our setup does not fully leverage RFA's potential.

Acknowledgments

This work was supported in part by NSF grant 2113530. Nikolaos Pappas was supported by the Swiss National Science Foundation grant P400P2_183911.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proc. of NAACL*.

M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and Marcello Federico. 2015. The IWSLT 2015 evaluation campaign. In *Proc. of IWSLT*. Downloaded from https://wit3.fbk.eu/2015-01.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proc. of ACL*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. of EMNLP*.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *Proc. of ICLR*.

Domenic Donato, Lei Yu, and Chris Dyer. 2021. Diverse pretrained context encodings improve document translation. In *Proc. of ACL*.

Denis Emelin, Ivan Titov, and Rico Sennrich. 2020. Detecting word sense disambiguation biases in machine translation for model-agnostic adversarial attacks. In *Proc. of EMNLP*.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. In *Proc. of NAACL*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation.

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. 2021. Finetuning pretrained transformers into RNNs. In *Proc. of EMNLP*.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proc. of the Fourth Workshop on Discourse in Machine Translation*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *Proc. of ICLR*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proc. of LREC*. Downloaded the processed version from `https://github.com/lena-voita/good-translation-wrong-in-context#cadec-data`.

António Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020. Document-level neural MT: A systematic comparison. In *Proc. of EAMT*.

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. 2021a. Luna: Linear unified nested attention. In *Proc. of NeurIPS*.

Zhiyi Ma, Sergey Edunov, and Michael Auli. 2021b. A comparison of approaches to document-level machine translation.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2021. A survey on document-level neural machine translation: Methods and evaluation. *ACM Comput. Surv.*, 54(2).

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proc. of EMNLP*.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proc. of WMT*.

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 *Proc. of NAACL*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proc. of WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. 2021. Random feature attention. In *Proc. of ICLR*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proc. of WMT*. Evaluation script at `https://github.com/mjpost/sacrebleu`.

Ali Rahimi and Benjamin Recht. 2008. Random features for large-scale kernel machines. In *Proc. of NeurIPS*.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *Proc. of ICML*.

Jürgen Schmidhuber. 1991. Learning to control fast-weight memories: An alternative to recurrent nets. Technical Report FKI-147-91, Institut für Informatik, Technische Universität München.

Jürgen Schmidhuber. 1992. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139.

Jürgen Schmidhuber. 1993. Reducing the ratio between learning complexity and number of time varying variables in fully recurrent nets. In *Proc. of ICANN*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. of ACL*.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proc. of the Third Workshop on Discourse in Machine Translation*.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to Remember Translation History with a Continuous Cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

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 *Proc. of NeurIPS*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves

on deixis, ellipsis, and lexical cohesion. In *Proc. of ACL*. Dataset and scoring script at https://github.com/lena-voita/good-translation-wrong-in-context.

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

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity.

Kayo Yin, Patrick Fernandes, Danish Pruthi, Aditi Chaudhary, André F. T. Martins, and Graham Neubig. 2021. Do context-aware translation models pay the right attention? In *Proc. of ACL*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proc. of EMNLP*.

Dataset	Lg.	Train Docs	Dev. Docs	Test Docs	Sent. /doc	Tok. /sent.
IWSLT	zh en	1713	8	56	121.5	20.4 22.6
Sub.	en ru	1.5M	10K	10K	4	10.3 9.5
Sub.-Cons.	en ru	—	2K	16K	4	10.5 9.6

Table 2: Dataset statistics of IWSLT, OpenSubtitles, and the consistency test sets for OpenSubtitles. We follow Ma et al. (2021b) in treating the four-sentence windows of OpenSubtitles as separate documents. The number of sentences per document and BPE tokens per sentence are averaged across all splits, except for OpenSubtitles-Consistency, which are only averaged across the development and test sets.

A Appendix

A.1 Dataset and Processing Details

The IWSLT 2015 dataset contains multilingual TED talk captions. Following Miculicich et al. (2018), we use the Chinese-to-English (zh-en) portion and use the *dev2010* subset for development and *tst2010-2013* for testing. We also use the processed OpenSubtitles2018 English-to-Russian (en-ru) dataset by Voita et al. (2019). The consistency test sets by Voita et al. (2019) measure (i) pronominal formality consistency (**deixis**), (ii) word choice consistency (**lexical cohesion**), (iii) inflection prediction accuracy of syntactically ambiguous words due to ellipsis (**ellipsis (inflection)**), and (iv) elided verb prediction accuracy (**ellipsis (VP)**). Models choose the candidate translation most consistent with the context and are scored with accuracy. We present an example taken from Voita et al. (2019) for lexical cohesion: the source English sequence is "Not for Julia. Julia has a taste for taunting her victims." and the target Russian translation candidates consist of two sequences, here transcribed with the Latin script: (a) "Ne dlya Dzhulii. Yuliya umeyet draznit' svoikh zhertv."; and (b) "Ne dlya Dzhulii. Dzhulii umeyet draznit' svoikh zhertv." The model is expected to choose (b) as it uses the same consistent translation for the name "Julia." Our random baseline randomly picks its translation from the candidate set. Table 2 summarizes dataset statistics.

We follow the tokenization of Miculicich et al. (2018). For all datasets, we first tokenize and true-case English and Russian with Moses (Koehn et al., 2007) and tokenize Chinese using Jieba.[4] We then run byte-pair encoding (Sennrich et al., 2016) on the concatenation of the training sets of the source and target languages using 30k splits, separately done for each dataset.

A.2 Hyperparameters and Training Details

Following Vaswani et al. (2017) and Peng et al. (2021), we use 6-layer transformers with 512 hidden dimension and 8 attention heads for both the encoder and decoder. Both RFA and the transformer baselines have 53M trainable parameters for IWSLT and 49M for OpenSubtitles, with the difference caused by different vocabulary sizes. We train all models in mixed-precision. We use the Adam optimizer (Kingma and Ba, 2015) with peak learning rate searched in $\{0.0005, 0.001\}$ warmed up through 8000 updates and an effective batch size of 16,384 in the number of tokens. We use beam size 4 for decoding. All other hyperparameters follow the recommendation in fairseq (Ott et al., 2019).[5] For RFA-sgate, to better enforce the inductive bias where sentences further away are less important, we treat the initialization of b_f in the sentential gating equation as a hyperparameter, searched in $\{1, 2\}$, instead of setting it to zero as in RFA. We search the RFA cross attention projection dimension + causal attention projection dimension in $\{128 + 64, 256 + 32\}$. We only employ gating in causal attention as we found it to hurt the performance when added in cross attention in preliminary experiments, degrading the performance by around 1 BLEU on IWSLT. According to Donato et al. (2021), source context is more important than target context, so it is possible that the model benefits from a non-decayed history on the source side.

We use early stopping with a patience of 10 epochs based on development set performance. Voita et al. (2019) observed that BLEU and consistency scores exhibit different training dynamics. We, therefore, train separate OpenSubtitles models when measuring BLEU versus consistency and use the respective metric for early stopping.

We manually tune the hyperparameters mentioned above based on the development set performance with the corresponding metric (i.e., BLEU or consistency). All final models use 0.001 learn-

[4] https://github.com/fxsjy/jieba

[5] https://github.com/pytorch/fairseq/tree/v0.10.0/examples/translation#iwslt14-german-to-english-transformer

L	1	2	3	4	5	10	15
B	1024	512	512	256	256	128	96

Table 3: The batch size (B) used to decode each window size L.

ing rate. The final IWSLT RFA models use $b_f = 2$ and RFA projection dimension $256 + 32$; OpenSubtitles (BLEU) RFA models use $b_f = 1$ and RFA projection dimension $256 + 32$; OpenSubtitles (consistency) RFA models use RFA projection dimension $128 + 64$.

We perform all training on a single NVIDIA 2080 Ti GPU. Depending on the dataset, window size, and model variant, each training run takes approximately 3.5 hours to a day.

A.3 Speedup Benchmark Details

In our synthetic benchmark setup, we decode under all L with the same trained model which allows us to examine the trend with a larger L. We use the smallest contexted model, $L = 2$, and verified that using it to decode in $L = 4$ yields a similar length distribution as an actual trained $L = 4$ model, confirming that this setup accurately reflects long context output length patterns. We benchmark with ungated RFA.[6] To simulate real-world usage, we use the largest possible batch size for each window size that fits on a single 32GB A100 GPU, the GPU that we use for all benchmark runs. In practice, as the transformer has larger memory consumption and since we use the same batch size between RFA and the transformer, this is the batch size that saturates the transformer. We only search the batch size over 2^k and 1.5×2^k for integer k for tractability. We report the batch sizes used in Table 3. The CPU experiments use the same batch sizes. We conduct this analysis on IWSLT as we believe OpenSubtitles represent a different genre from many settings where long contexts are expected to be useful, though in this synthetic setup, the trend would be similar when the sequence length is controlled. The CPU experiments are run with six 2.2GHz Intel Cascade Lake CPUs.

[6]The speed difference between the RFA variants is negligible as gating requires minimal additional computation. This is also confirmed by Peng et al. (2021), where their per-token gating has the same speedup as no gating.