Modeling Context With Linear Attention for Scalable Document-Level Translation

Anonymous ACL submission

Abstract

Document-level neural machine translation allows models to leverage dependencies beyond sentence-internal context to produce more coherent and consistent translations. However, these models, predominantly based on transformers, are difficult to scale to long documents due to the quadratic time and space complexity of their self-attention layers. Recent efforts on efficient attention variants improve scalability, but it is yet unclear if and to what extent their inductive biases are suitable for document translation. In this paper, we explore the efficacy of a recent linear attention model by Peng et al. (2021) on document-level translation and augment it with a sentential gating mechanism. We evaluate the model on the IWSLT 2015 and OpenSubtitles 2018 datasets against a strong transformer baseline and achieve up to 40% decoding speedup with similar or improved BLEU scores. We show that the sentential gate further improves translation quality on IWSLT, a dataset with long sequences.

1 Introduction

Sentence-level neural machine translation has seen significant recent progress (Bahdanau et al., 2015; Vaswani et al., 2017). Document-level translation facilitates a more general version of translation when inter-sentential context is accessible, such as paragraphs, documents, or 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 machine translation (Vaswani et al., 2017; Chen et al., 2018; Wang et al., 2019) and have become the default architecture for document translation. However, they do not scale well in the sequence length due to the quadratic complexity of self-attention and hence can be computationally prohibitive to translate long text. Alternative architectures exist, but most are still quadratic in the context length (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).

Recent efficient self-attention variants reduce complexity (Guo et al., 2019; Child et al., 2019; Kitaev et al., 2020; Wang et al., 2020, *i.a.*), though many do not focus on decoding speed. Random feature attention (RFA; Peng et al., 2021) admits a recurrent computation, suitable for autoregressive generation. With few extra parameters, it approximates softmax attention in linear time and space and has proved successful in machine translation. However, it has not been tested on document translation where its asymptotic improvement is expected to bring large efficiency gains. In this work, we investigate its effectiveness on document translation and achieve up to 40% speedup with similar BLEU, or sometimes improved BLEU on long sequences. We also equip RFA with a sentential gate, bringing inductive biases tailored to representing document context for machine translation.

Our main contributions are: (i) we study the efficacy of RFA for document translation; (ii) we incorporate a sentential gating mechanism into RFA tailored to document translation; (iii) we experimentally validate that RFA is competitive with transformer and up to 40% faster on document translation. Our proposed gating model yields the best performance in BLEU for long sequences. To encourage research on scalable document-level translation, we will release our code upon publication.

2 Background

Standard machine translation independently translates each source sentence into the target. However, translating sentence-by-sentence discards useful context information that can assist lexical choice

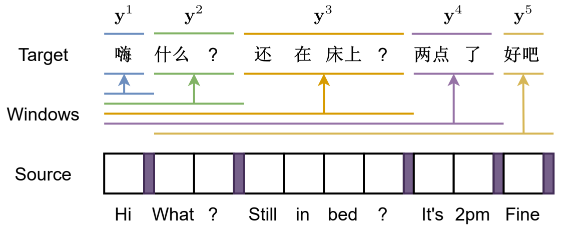

Figure 1: The concatenation model for document translation with a sliding window of length $L = 4$. Every window is translated in its entirety, but only the last translated sentence is used for evaluation. The purple bars denote the sentence separator token.

and ambiguity resolution. Document-level translation further conditions on previous source and target sentences. It respects document context and preserves sentence interaction to produce more coherent translations (Voita et al., 2019).

The Concatenation Model. Many document-contextual models complicate the transformer architecture (Miculicich et al., 2018; Donato et al., 2021, *i.a.*). Recent studies have shown that the simple concatenation model that directly translates the source document (or a multi-sentence window) to the target document with a single encoder-decoder stack 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. We adopt this model in this work, though it has poor scalability, which we explain next.

Scalability of Self-Attention. Transformers contain three types of attention layers: 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:

$$\operatorname{attn}(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}) = \sum_{i=1}^{N} \frac{\exp(\mathbf{q}_t \cdot \mathbf{k}_i)}{\sum_{i'=1}^{N} \exp(\mathbf{q}_t \cdot \mathbf{k}_{i'})} \mathbf{v}_i^\top$$

where N is the sequence length. This pairwise interaction consumes quadratic time and memory 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).

3 Scalable Document-Level Translation

For the first time, we test random feature attention, which has demonstrated efficiency in autoregressive decoding, as a linear time and space model to improve the scalability of document translation.[1] We also augment it with a sentential gate to circumvent capacity constraints with a long context.

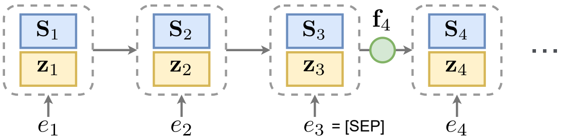

Figure 2: Our sentential gating mechanism. e_1 and e_4 are at the beginning of two sentences.

3.1 Random Feature Attention

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

$$\textsc{Rfa}(\mathbf{q}_t, \{\mathbf{k}_i\}, \{\mathbf{v}_i\}) = \frac{\boldsymbol{\phi}(\mathbf{q}_t) \cdot \mathbf{S}_t}{\boldsymbol{\phi}(\mathbf{q}_t) \cdot \mathbf{z}_t}$$

where $\boldsymbol{\phi}(\cdot)$ is a random nonlinear transformation where $\boldsymbol{\phi}(\mathbf{q}) \cdot \boldsymbol{\phi}(\mathbf{k}) \approx \exp \mathbf{q} \cdot \mathbf{k}$ in expectation over $\boldsymbol{\phi}$ (Rahimi and Recht, 2008); $\mathbf{S}$, $\mathbf{z}$ summarize the 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 sentence. In cross attention, $\mathbf{S}$ and $\mathbf{z}$ represent the source sentence and are constant for all query positions t: $\mathbf{S}_t = \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{\phi}(\mathbf{k}_i)\mathbf{v}_i^\top$ and $\mathbf{z}_t = \sum_{i=1}^{|\mathbf{x}|} \boldsymbol{\phi}(\mathbf{k}_i)$. In causal attention, they represent the target prefix $i \leq t$: $\mathbf{S}_t = \sum_{i=1}^{t} \boldsymbol{\phi}(\mathbf{k}_i)\mathbf{v}_i^\top = \mathbf{S}_{t-1} + \boldsymbol{\phi}(\mathbf{k}_t)\mathbf{v}_t^\top$ and $\mathbf{z}_t = \sum_{i=1}^{t} \boldsymbol{\phi}(\mathbf{k}_i) = \mathbf{z}_{t-1} + \boldsymbol{\phi}(\mathbf{k}_t)$. 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 step. RFA serves as a drop-in replacement for attn 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 a complete discussion of 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, such as as exposed 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.

To address this, inspired by gated RNNs (Cho et al., 2014, *i.a.*), we augment RFA with a sentence-

[1] RFA was the first model to demonstrate decoding speed improvements in translation, making it appropriate for this study. Many other linear attention models have been proposed since this work was carried out (Kasai et al., 2021; Schlag et al., 2021; Ma et al., 2021a), and it would be exciting future work to investigate their utility in document translation.

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 a word x_t with representation $\mathbf{e}_t$, we compute a forget gate using the separator token between sentences:

$$f_t = \begin{cases} \sigma(\mathbf{w}_f \cdot \mathbf{e}_{t-1} + b_f) & \text{if } x_t \text{ starts a sentence} \\ 1 & \text{otherwise} \end{cases}$$
$$\mathbf{S}_t = f_t\, \mathbf{S}_{t-1} + \boldsymbol{\phi}\left(\mathbf{k}_t\right) \mathbf{v}_t^\top$$
$$\mathbf{z}_t = f_t\, \mathbf{z}_{t-1} + \boldsymbol{\phi}\left(\mathbf{k}_t\right)$$

where σ denotes the sigmoid function. 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 enforces an inductive bias that, intuitively, previous sentences are less important in translation, and their representations are exponentially decayed.

Relation to Prior Work. While gating is common in RNNs, it is less clear how it applies to transformers. Miculicich et al. (2018) gate 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).[2] To quantify discourse consistency, we also use the test sets by Voita et al. (2019) that are based on OpenSubtitles. We introduce these datasets in more detail in Appendix 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 $L = 4$. During inference, we use the last translated sentence in each window for evaluation. For a more granular analysis, we consider $L \in [1, 4]$ for consistency experiments. More details are in Appendix A.1.

Model Settings. We compare RFA and transformer with the concatenation model. For RFA, we experiment with the ungated (RFA) and sentential-gated (RFA-sgate) versions. 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-balanced). We mostly default to fairseq hyperparameters (Ott et al., 2019), most suitable for the $L = 1$ transformer (see Appendix A.2). We measure decoding speed in the number of decoded tokens over the forward pass time. We do not benchmark with OpenSubtitles as its short sequences (≈ 10 tokens per sentence; see Table 2, appendix) are not expected to show a speedup. We believe movie subtitles represent a different genre from many settings where long contexts are expected to be useful. We follow Ott et al. (2018) and cache previous $\mathbf{k}$ and $\mathbf{v}$ for our *baseline* which substantially increases its speed.

5 Results

Speed. Table 1 (top) shows the speedup of the ungated RFA over transformer.[3] RFA offers a considerable speedup, consistent across both window sizes. This is especially pronounced at $L = 4$ due to RFA's linear complexity. This makes RFA an attractive choice since, as demonstrated below, models with longer context are the best at capturing discourse phenomena. In particular, we only experimented with window size up to 4 in this work, limited by the dataset design of Voita et al. (2019). In reality, however, RFA can be combined with an even longer context to capture longer-range dependency and offer a more prominent speedup.

We note that all decoding is done on GPUs. If performed on TPUs, as was done in Peng et al. (2021), the feed-forward layers would be much faster, and the attention layers would take a larger fraction of the decoding time. This would make the RFA speedup more pronounced. For comparison,

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

[3] 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.

		IWSLT		Subtitles	
	Window Size L	1	4	1	4
Speed	Transformer	150	36	—	—
	RFA	179	49	—	—
	Speedup	1.2×	1.4×	—	—
Quality	Transformer	31.7	30.4	32.6	33.1
	RFA	31.0	**30.7**	**32.9**	**33.2**
	RFA-sgate-balanced	—	**30.8**	—	33.0
	RFA-sgate	—	**31.2**	—	**33.2**

Table 1: Inference speed, in the number of decoded tokens / second, and BLEU on IWSLT and OpenSubtitles test sets. Bold scores outperform transformer. Our baselines are optimized: see Appendix A.3 for comparison with prior work. We do not use batch decoding as it is non-trivial with sliding windows, and we expect it would help the speed similarly for both models.

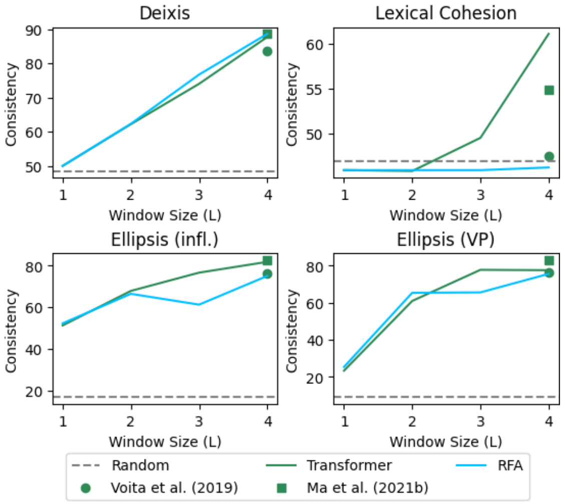

Figure 3: Model performance on the consistency test set, broken down into phenomena. Transformer and RFA are tested with window sizes from 1 to 4. We compare with the baselines in Voita et al. (2019) and Ma et al. (2021b) corresponding to our Transformer $L = 4$.

Peng et al. (2021) reported 1.8–1.9× speedup for single sentence decoding compared to our 1.2×.

BLEU Score. Table 1 (bottom) shows BLEU scores on IWSLT and OpenSubtitles. Overall, RFA performs slightly better than transformer. The only exception is the high IWSLT performance of the sentence-level transformer, which could be due to defaulting to fairseq hyperparameters that are designed for this setting. The gated RFA model is the best on IWSLT at $L = 4$, demonstrating 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 sentences (see Table 2, appendix), gating is less useful on this dataset. Our gate also outperforms the balanced variant in Miculicich et al. (2018) and Peng et al. (2021), showing its better suitability for document translation. Similar to previous work (Voita et al., 2019; Ma et al., 2021b), longer context does not clearly lead to better BLEU scores, though it improves consistency metrics, to which we turn next.

Discourse Consistency Scores. Figure 3 plots the consistency scores in four phenomena for RFA and transformer. As gating is not helpful for OpenSubtitles in BLEU, we only compare with ungated RFA. We also compare to a random baseline and the concatenation models from Voita et al. (2019) and Ma et al. (2021b), conceptually the same as our $L = 4$ transformer, though with unavoidable implementation discrepancies that explain their performance differences. Though it is not clear for BLEU, longer context almost monotonically yields better consistency scores. This highlights the benefit of translating with longer context, a setting where RFA achieves better speedup, shown above.

RFA slightly underperforms transformer in most settings. We hypothesize that the direct query-key interaction in softmax attention is more suitable for precise long-distance information extraction, usually required for consistency metrics, than the RFA approximation. RFA is not able to learn lexical cohesion, comparing to the random baseline. This is also the case for Voita et al. (2019)'s baseline. And while Ma et al. (2021b)'s performs better, it is still much worse than our transformer baseline. Zhang et al. (2020) also noted this difficulty whose proposed method also underperforms our random baseline. They used the Partial Copy mechanism (Jean et al., 2019) as a remedy, though at the expense of other metrics. This is orthogonal to our approach. Our results reveal that, while efficient transformers may provide an attractive speedup while retaining or improving some automatic evaluation scores, they may do worse on other metrics. We, therefore, call for a more holistic evaluation of these models to fully understand all the trade-offs.

6 Conclusion

We explored the effectiveness of random feature attention, combined with sentential gating, on document translation. We demonstrated that our model provides a speedup over transformer by up to 40% with similar BLEU scores. Our sentential gate also proves effective, especially on long sequences.

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. *CoRR*, abs/1904.10509.

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*.

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

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

Sébastien Jean, Ankur Bapna, and Orhan Firat. 2019. Fill in the blanks: Imputing missing sentences for larger-context neural machine translation.

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.

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.

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 *International Conference on Artificial Neural Networks (ICANN)*, pages 460–463, Amsterdam, Netherlands.

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.

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*.

Pei Zhang, Boxing Chen, Niyu Ge, and Kai Fan. 2020. Long-short term masking transformer: A simple but effective baseline for document-level neural machine translation. In *Proc. of EMNLP*.

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

We follow the tokenization of Miculicich et al. (2018). For all datasets, we first tokenize and truecase 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 baseline 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.

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 learning 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 and decoding on a single NVIDIA 2080 Ti GPU.

A.3 Comparison to Previous Work

We note that our transformer baseline model in Table 1 is very optimized. We offer a few points of reference in this section, though the numbers are not directly comparable as we used SacreBLEU (Post, 2018), which offers a standard BLEU computation

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

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

and enables better comparability across research works, while they did not. Also, Miculicich et al. (2018) evaluated on a sentence basis while we evaluated document-level BLEU following Ma et al. (2021b). On IWSLT, our baseline (31.7 BLEU) outperforms the baseline reported in Miculicich et al. (2018) with 16.87 BLEU when $L = 1$. On OpenSubtitles, our baselines also outperform the ones in Voita et al. (2019) which achieved 32.40 ($L = 1$) and 31.56 ($L = 4$) BLEU, compared to our 32.6 and 33.1.