Sequence Shortening for Context-Aware Machine Translation

Anonymous ACL submission

Abstract

Context-aware Machine Translation aims to improve translations of sentences by incorporating surrounding sentences as context. Towards this task, two main architectures have been applied, namely single-encoder (based on concatenation) and multi-encoder models. In this study, we show that a special case of multi-encoder architecture, where the latent representation of the source sentence is cached and reused as the context in the next step, achieves higher accuracy on the contrastive datasets (where the models have to rank the correct translation among the provided sentences) and comparable COMET scores as the single- and multi-encoder approaches. Furthermore, we investigate the application of Sequence Shortening to the cached representations. We test three pooling-based shortening techniques and introduce two novel methods - Latent Grouping and Latent Selecting, where the network learns to group tokens or selects the tokens to be cached as context. Our experiments show that the two methods achieve competitive BLEU scores and accuracies on the contrastive datasets to the other tested methods while potentially allowing for higher interpretability and reducing the growth of memory requirements with increased context size.

1 Introduction

Following the introduction of the Transformer model (Vaswani et al., 2017), Sentence-level Machine Translation, where the task is to translate separate sentences, has seen great success in recent years (Vaswani et al., 2017; Hassan et al., 2018; Costa-jussà et al., 2022; Tiedemann et al., 2022). However, real-world applications of the translation systems are often used to translate a whole document or a longer discourse (e.g. a transcribed speech). In those circumstances, Sentence-level Machine Translation processes each sentence separately and is incapable of leveraging the surrounding or previous sentences (referred to as the context sentences). This is in contrast to the Context-aware Machine Translation where the context sentences are available to the system. The information in the previous sentences can be helpful to maintain the coherence of the translation and to resolve ambiguities (Agrawal et al., 2018; Bawden et al., 2018; Müller et al., 2018; Voita et al., 2019b). Both the sentences of the text in the source language and the previously translated sentences can be used as context. The former is referred to as source-side context and the latter as target-side context.

Many Context-aware Machine Translation approaches have been proposed including novel architectures that can be broadly categorized into *single-encoder* and *multi-encoder* types. In single-encoder architectures, the context sentences are concatenated with the current sentence and processed as a long sequence by a single encoder (Tiedemann and Scherrer, 2017; Agrawal et al., 2018; Ma et al., 2020; Zhang et al., 2020; Majumde et al., 2022). In multi-encoder architectures, the context sentences are processed by a separate encoder than the current sentence (Tu et al., 2017; Bawden et al., 2018; Miculicich et al., 2018; Maruf et al., 2019; Huo et al., 2020; Zheng et al., 2021). Several multi-encoder approaches (Voita et al., 2018; Li et al., 2020) involve sharing parameters of encoders. This approach reduces the number of parameters and could also increase the speed of translation when translating the whole document sentence-by-sentence. Inspired by this idea, we investigate multi-encoder architectures where all the encoder parameters are shared (Tu et al., 2018; Voita et al., 2019b; Wu et al., 2022), which allows caching the hidden representation of the current sentence and reusing it as the hidden representation of the context when translating subsequent sentences. In this study, we refer to this architecture as *caching*.

In Transformers, the number of tokens does not change during the processing of the sequence

through the encoder (and decoder) layers. Concurrent to Machine Translation, several techniques have been proposed to shorten the sequence of tokens in the task of language modeling (Subramanian et al., 2020; Dai et al., 2020; Nawrot et al., 2022). In particular, the tokens are combined in the shortening modules that are added between a specified number of encoder layers. Sequence Shortening can lead to the reduction of the computational and memory requirements in the subsequent layers as the requirements of the self-attention module grow quadratically with the number of tokens (although a substantial amount of research is done to mitigate that (Kitaev et al., 2020; Wang et al., 2020)).

In this paper, we investigate the application of Sequence Shortening to Context-aware Machine Translation. Specifically, we apply the shortening of the cached hidden representations of the context sentences in the caching multi-encoder architectures. The intuition behind this approach is that a compressed representation of the previously seen sentences should be enough to use as a context while possibly decreasing the computational and memory requirements during inference. Sequence Shortening can be seen as related to the concept of *chunking* from psychology (Miller, 1956; Terrace, 2002; Mathy and Feldman, 2012). To limit the scope, we consider only the source-side context. Additionally, we introduce *Latent Grouping* and *Latent Selecting* - new shortening techniques where the network can learn how to group or select tokens to form a shortened sequence.

2 Related Work

2.1 Context-aware Machine Translation

A straightforward approach to incorporate context into Machine Translation is to concatenate previous sentences with the current sentence, which has been referred to as *concatenation* or *single-encoder* architecture because only a single encoder is used (Tiedemann and Scherrer, 2017; Ma et al., 2020; Zhang et al., 2020; Majumde et al., 2022). The *multi-encoder* approach is to encode the context sentences by a separate encoder (Jean et al., 2017; Miculicich et al., 2018; Maruf et al., 2019; Huo et al., 2020; Zheng et al., 2021). While the encoders are separate in multi-encoder architectures, weight-sharing between them has been investigated in previous works (Voita et al., 2018; Tu et al., 2018; Li et al., 2020; Wu et al., 2022). Existing studies also investigated hierarchical attention (Miculicich et al., 2018; Bawden et al., 2018; Wu et al., 2022; Chen et al., 2022), sparse attention (Maruf et al., 2019; Bao et al., 2021), aggregating the hidden representation of the context tokens (Morishita et al., 2021), post-processing the translation (Voita et al., 2019b,a), and using a memory mechanism (Tu et al., 2018; Feng et al., 2022).

Mostly orthogonal to architectural approaches, another line of work concentrates on making the models use the context more effectively. These methods utilize regularization such as dropout of the tokens in the source sentence (CoWord dropout; Fernandes et al., 2021) or attention regularization based on human translators (Yin et al., 2021) and data augmentation (Lupo et al., 2022) along with contrastive learning (Hwang et al., 2021).

It has been argued that widely used sentence-level metrics (such as BLEU (Papineni et al., 2002)) are ill-equipped to measure the translation quality with regard to the inter-sentential phenomena (Hardmeier, 2012; Wong and Kit, 2012). For this reason, research has been done to measure the usage of context by machine translation models, where two main avenues have been explored: introducing new metrics (Fernandes et al., 2021, 2023) and contrastive datasets (Müller et al., 2018; Bawden et al., 2018; Voita et al., 2019b; Lopes et al., 2020). In the contrastive datasets, the model is presented with the task of ranking several translations of the same source sentence with the same context. The provided translations differ only partially and the provided context is required to choose the correct translation.

2.2 Sequence Shortening

Sequence Shortening has been introduced as a way to exploit the hierarchical structure of language to reduce the memory and computational cost of the Transformer architecture (Subramanian et al., 2020; Dai et al., 2020; Nawrot et al., 2022). Shortening can be done by average pooling of the hidden representation of the tokens (Subramanian et al., 2020). Allowing the tokens of the shortened sequence to attend to the hidden representation of the original sequence was found beneficial (Dai et al., 2020). Replacing average pooling with the linear transformation of the concatenated representation of the tokens of the original sequence has also been used (Nawrot et al., 2022). Another way of shortening the sequence is to find and retain only the most

important tokens of the original sequence (Goyal et al., 2020).

The most related work to one of our methods *Latent Grouping* is the Charformer (Tay et al., 2021) architecture, where the tokenization is performed by a sub-network that learns to select block sizes for characters of the input sequence. The block size representations are subsequently summed with weights predicted by the sub-network. *Latent Grouping* differs from Charformer in the placement of the grouping (after the encoder in the case of *Latent Grouping*) and the aggregated representation (encoder representations of tokens themselves in the case of *Latent Grouping*).

Our work lies in the intersection of Context-aware Machine Translation and Sequence Shortening. We test the performance of caching architecture against single- and multi-encoder architectures and investigate applying shortening to the cached sentences.

3 Background

3.1 Transformer

The Transformer architecture, introduced for sentence-level translation, consists of the encoder and decoder (Vaswani et al., 2017). The sentence in the source language is tokenized and embedded before it is passed to the encoder. The encoder processes the sequence by L consecutive encoder layers, each consisting of the self-attention module and the element-wise feed-forward network. Residual connection is added around both modules followed by Layer Normalization (Ba et al., 2016).

Hidden representation of the L-th encoder layer H^L is fed into the decoder, which auto-regressively produces the output sequence $Y = (y_1, ..., y_T)$, until it reaches the end-of-sequence token. Decoder layers process the currently produced sequence with the self-attention module, followed by the cross-attention module and feed-forward network. Unlike in the encoder, the self-attention module in the decoder uses causal masking (the tokens can not attend to the future tokens). In Cross-attention, multi-head attention uses the decoded sequence as queries and encoder output as keys and values. Residual connection and Layer Normalization are applied after each module.

3.2 Pooling-based Shortening

Sequence Shortening is a method that results in a reduction in the number of tokens in a sequence by combining the tokens of the hidden representation of the input sequence H^L. In the pooling-based shortening the sequence is divided into non-overlapping groups of K neighboring tokens each (K is a hyper-parameter). Pooling of the tokens in each group is then performed:

$$\tilde{G} = \text{Pooling}(H^L), \tag{1}$$

where $\tilde{G}$ is the sequence of size $\lceil M/K \rceil$ of the pooled tokens. Subsequently, the pooled tokens $\tilde{G}$ attend to the hidden representation of the original sequence using the attention module followed by the residual connection and the Layer Normalization:

$$G = \text{LayerNorm}(\tilde{G} + \text{Attn}(\tilde{G}, H^L, H^L)), \tag{2}$$

where G is the final shortened sequence. Commonly used pooling operations are average (Dai et al., 2020) and linear pooling (Nawrot et al., 2022) (transformation of the concatenated tokens).

4 Method

4.1 Latent Grouping

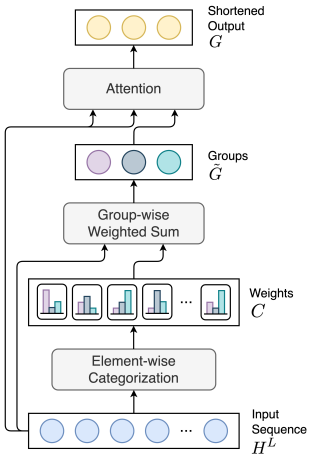


Figure 1: Illustration of Grouping Shortening with the number of groups set to three.

In contrast to pooling, Latent Grouping, illustrated in Figure 1, results in a fixed number of tokens in the shortened sequence corresponding to the number of groups K, which is a hyper-parameter. Each token is categorized into a group by the feed-forward network with the number of outputs equal to the number of groups. We obtain the categorization for the i-th token to k-th group $c_{i,k}$ by applying the Softmax function to the outputs

in the dimension of the groups:

$$\mathbf{c}_i = \text{Softmax}(\text{FFN}(\mathbf{h}_i^L)), \quad \forall i = 1, ..., M, \tag{3}$$

where $\mathbf{h}^L$ is the hidden representation of the last encoder layer and $\mathbf{c}_i$ is the vector of size K representing the categorizations of the i-th token to all the groups. As an alternative to Softmax, Sparsemax function (Martins and Astudillo, 2016) can also be used resulting in the categorizations of tokens that are more sparse, which means that a token is categorized into a smaller number of groups, and most categorizations are equal to zero. Subsequently, the groups $\tilde{G}$ are constructed as the sum of the hidden representations $\mathbf{h}^L$ with categorizations $a_{i,k}$ used as weights:

$$\tilde{g}_k = \sum_i c_{i,k} h_i^L, \quad \forall k = 1, ..., K, \tag{4}$$

where $\tilde{g}_k$ is a k-th grouped token composing the sequence $\tilde{G}$ in the equation (1). The network learns how to soft assign each token to the groups. A group representation is computed using the weighted average of tokens, which makes back-propagations into the categorizing network possible. Finally, the attention module is applied as in equation (2).

4.2 Latent Selecting

Latent Selecting differs from Latent Grouping by enabling the groups to select tokens to aggregate rather than assigning each token to a group and allowing the model to ignore tokens entirely rather than assigning them to at least one group. This is similar to selecting the *hub* tokens in Power-BERT (Goyal et al., 2020), where the selection is based on the attention scores of the previous layer. Although Latent Selecting can be achieved by maintaining a categorizing feed-forward network for each group, we utilize the same network as described for Latent Grouping but apply the Softmax (or Sparsemax) function in equation (3) in the sequence dimension instead of the token dimension.

4.3 Context Shortening

The architecture we use, illustrated in Figure 2, is based on caching the hidden representations produced by the encoder, where the representations of the tokens of the current sentence are stored and can be reused as context when the subsequent sentences are translated. Although this architecture uses only a single encoder, it is different from the single-encoder models because the current sentence and the context sentences are processed separately. While in the standard caching architecture the hidden representation of all the tokens is stored, we introduce a Sequence Shortening module directly after the encoder, which returns the compressed hidden representation usually containing fewer tokens than the original sequence. We consider: mean pooling (Dai et al., 2020), max pooling, linear pooling (Nawrot et al., 2022), latent grouping, and latent selecting. Additionally, we also test the simple aggregation of the whole context sequences into a single vector by averaging the tokens. Conceptually, Sequence Shortening of the context can be seen as a middle-ground between storing tokens and sentence aggregations.

The integration of the context with the decoder can also be done in several ways. Firstly, the context sentences can be concatenated to the current sentence. This method is similar to the single-encoder (concatenating) architecture, where the difference is that the encoder does not have access to other sentences in the case of caching architecture. In this case, the decoder layers are the same as in the vanilla transformer with self- and cross-attention modules. Secondly, the context sentences can be processed in the decoder layers by a separate context-attention module, where the decoder tokens attend to the context tokens. We experiment with the parallel and serial alignment of the cross- and context-attention modules. Additionally, we also experiment with gating the representation resulting from applying context-attention using the following equation:

$$\lambda_i = \sigma(\text{FFN}(\hat{\mathbf{h}}_i)), \quad \hat{\mathbf{h}}'_i = \lambda_i \hat{\mathbf{h}}_i, \quad \forall i = 1, ..., M \tag{5}$$

where $\hat{\mathbf{h}}_i$ is the i-th token representation returned by the context-attention module, FFN is a token-wise linear layer with one output, σ is the Sigmoid function.

For Sentence Aggregation and Shortening architectures, the aggregated or shortened representation of the current sentence can be included in context sentences. This helps with the training, as often none of the previous sentences has an effect on the translation, known as the two-fold sparsity problem (Lupo et al., 2022), and the context attention

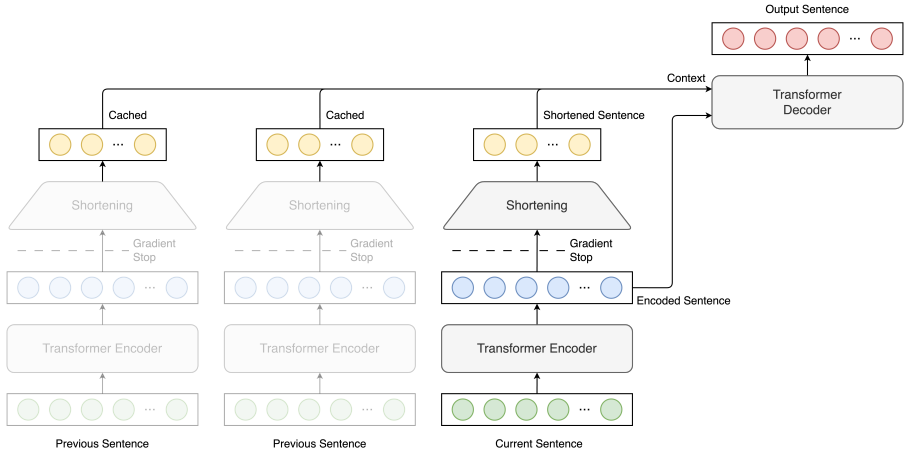


Figure 2: The illustration of a Shortening Architecture with the representation of the two previous sentences being cached. The dashed line represents the optional blocking of the gradient during training.

module can still be trained to attend to the representation of the current sentence. To allow the decoder to distinguish between context sentences we employ learned segment embeddings (Devlin et al., 2019). Similarly, we also add learned positional encoding for the shortened tokens inside context sentences.

During training, caching is not used, meaning that the model receives tokenized context sentences and processes them using the same encoder. This implies that the weights of the encoder receive the backpropagation gradient from multiple sources - the current sentence and each of the context sentences, which can lead to difficulties in training. Therefore, we consider blocking the gradient after the encoder and before shortening where applicable by allowing the gradient information to flow for a specified number of context sentences, after which, the gradient is blocked.

5 Experiments

All our experiments are implemented[1] in *fairseq* framework (Ott et al., 2019). We used the code repository of Fernandes et al. (2021) as the base for our implementation.

5.1 Data

We used the English to German and English to French directions of the IWSLT 2017 (Cettolo et al., 2017) document-level dataset that is based on the subtitles of the TED Talks[2]. Following Fernandes et al. (2021), we used *tst2011-tst2014* as validation subset and *tst2015* as the test subset. The data is byte-pair encoded (Sennrich et al., 2016) using SentencePiece framework (Kudo and Richardson, 2018) on the training subset with 20,000 vocabulary size for each language separately (see Table 1). We measured BLEU (Papineni et al., 2002) using *sacreBleu* library (Post, 2018). We also report COMET (Rei et al., 2020) in Appendix B.

Dataset	Docs	Sent/Doc	Tok/Sent
En-De Train	1698	121.4	21.9
En-De Valid	62	87.6	20.6
En-De Test	12	90.0	20.8
En-Fr Train	1914	121.6	22.0
En-Fr Valid	66	88.2	20.9
En-Fr Test	12	100.8	21.4

Table 1: The details of the IWSLT 2017 datasets.

To measure the context usage of the trained models, we employed ContraPro (Müller et al., 2018) contrastive dataset for English to German direction, and the contrastive dataset by Lopes et al. (2020) for English to French direction. Both are based on the OpenSubtitles 2018 dataset (Lison et al., 2018). These datasets consist of the source sentence with the context (previous sentences on the source and target side) with several translations differing only in a pronoun that requires context to be correctly translated. Models rank the translations by assigning probabilities to each of them. The translation is considered to be accurate when the right translation

[1]The code for this paper (based on https://github.com/neulab/contextual-mt) can be found on Github https://anonymous.4open.science/r/shortening-context-mt-F8C1.

[2]https://www.ted.com/

Model	BLEU	Accuracy				
Sentence-level	28.11	43.67%				
	Context: 1		Context: 2		Context: 3	
Model	**BLEU**	**Accuracy**	**BLEU**	**Accuracy**	**BLEU**	**Accuracy**
Single-encoder	28.31	47.42%	27.95	48.18%	27.88	48.88%
Multi-encoder	**28.67**	44.93%	28.50	46.65%	28.26	45.00%
Caching Tokens	28.35	54.06%	28.50	54.13%	**29.08**	51.23%
Caching Sentence	28.38	45.72%	26.73	45.26%	26.70	44.91%
Shortening - Max Pooling	27.62	51.67%	27.88	**55.08%**	28.26	50.89%
Shortening - Avg Pooling	28.09	53.37%	27.85	54.81%	28.38	50.54%
Shortening - Linear Pooling	27.62	52.71%	28.03	52.13%	28.18	51.27%
Shortening - Grouping	28.21	**56.98%**	**28.70**	54.51%	28.49	51.16%
Shortening - Selecting	28.15	54.48%	28.55	54.21%	28.01	**51.95%**

Table 2: Results of the **En-De** IWSLT 2017 experiment. The models were trained to use only the source-side context. We report BLEU of the test subset and the accuracy of the ContraPro (Müller et al., 2018) contrastive dataset.

is ranked the highest by the model.

5.2 Models

Based on the described methods, we trained the following caching models:

- **Caching Tokens** - where the encoder representations of the context sentences are stored directly,
- **Caching Sentences** - where the representations of the context sentences are averaged and stored,
- **Shortening - Mean Pooling** - Sequence shortening with Mean Pooling applied to the outputs of the encoder, based on (Dai et al., 2020),
- **Shortening - Max Pooling** - shortening with Max Pooling,
- **Shortening - Linear Pooling** - shortening with Linear Pooling, based on (Nawrot et al., 2022),
- **Shortening - Grouping** - shortening with Latent Grouping (Section 4.1),
- **Shortening - Selecting** - shortening with Latent Selecting (Section 4.2).

For all the aggregating models, the current sentence is also used as context and is concatenated with the context sentences after embedding. Moreover, we also test the following baseline models:

- **Sentence-level Transformer** - where context sentences are ignored,
- **Single-encoder Transformer** - where context sentences are prepended to the current sentence and processed by the encoder, we used Fernandes et al. (2021) implementation,
- **Multi-encoder Transformer** - with the separate encoder (without weights-sharing) used to encode the context sentences, again based on the Fernandes et al. (2021) implementation, where the context and the current sentence are concatenated in the decoder. Our experiments revealed that this integration yields better results than with the separate context-attention module.

All tested models are based on the Transformer base architecture (Vaswani et al., 2017). The hyper-parameters and model details can be found in Appendix A. We tuned the hyper-parameters of the models based on the performance on the validation subset. From the K values of $[2, 3, 4]$ for pooling architectures 2 was selected. For grouping and selecting architectures, we considered K values of $[8, 9, 10, 11]$ and selected 9 and 10 respectively for English-German translation and 11 (for both models) for English-French translation. For the categorizing network, we used one hidden layer with 512 units and the Sparsemax activation function to obtain more sparse categorizations. We performed preliminary experiments to find the architectural choices (gradient stopping and decoder integration) for each caching model. In Caching Tokens, Caching Sentence, and Pooling architectures, we block gradient past the encoder for context sentences. Additionally, we allow gradient into the shortening from one and two context sentences for Selecting and Grouping architectures respectively. All models apart from Caching Sentence use sequential attention modules in the decoder (self-attention, cross-attention, and context-attention)

Model	BLEU	Accuracy				
Sentence-level	37.64	75.92%				
	Context: 1		Context: 2		Context: 3	
Model	**BLEU**	**Accuracy**	**BLEU**	**Accuracy**	**BLEU**	**Accuracy**
Single-encoder	37.25	77.27%	37.18	78.98%	37.12	**80.87%**
Multi-encoder	37.44	75.72%	37.12	77.23%	37.34	75.76%
Caching Tokens	36.88	79.67%	37.29	80.14%	37.73	79.90%
Caching Sentence	36.50	77.33%	34.21	76.25%	34.78	75.71%
Shortening - Max Pooling	**37.48**	79.51%	36.72	80.59%	37.85	79.71%
Shortening - Avg Pooling	37.13	77.75%	37.12	80.16%	**38.18**	80.41%
Shortening - Linear Pooling	37.02	80.47%	37.12	79.37%	37.42	79.64%
Shortening - Grouping	37.05	79.91%	**37.98**	**81.13%**	37.18	79.54%
Shortening - Selecting	37.38	**80.89%**	37.83	80.32%	37.81	80.09%

Table 3: Results of the **En-Fr** IWSLT 2017 experiment. The models were trained to use only the source-side context. We report BLEU of the test subset and the accuracy of the contrastive dataset by Lopes et al. (2020).

without any gating mechanism. Caching Sentence yields the highest performance when parallel cross- and context-attention decoder is used with the gate on the context branch (see equation (5)).

5.3 Results

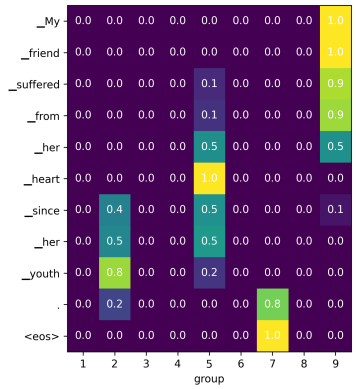
The results of the single run (with the predetermined seed) of the English-to-German translation on the IWSLT 2017 dataset can be seen in Table 2. The BLEU score of the context-aware models is generally similar to or slightly higher than the sentence-level Transformer. BLEU does not correlate well with the contrastive accuracy, which is strictly higher for all context-aware models. This confirms that sentence-level metrics do not reflect the context usage of the models. The highest contrastive dataset accuracy was achieved by the Grouping Shortening model for the context size of one, the Max Pooling Shortening model for the context size of two, and the Selecting Shortening model for the context size of three. The highest accuracy averaged over the tested context sizes was reached by the model employing Latent Grouping, followed by the Latent Selecting model. Caching Tokens architecture exhibits comparable COMET scores to the Single- and Multi-encoder architectures while achieving higher accuracy on the contrastive dataset. Caching Sentence architecture performed worse than other tested models, suggesting that representing the whole sentence as a single vector is not sufficient for contextual translation.

Table 3 shows the results of the English-to-French translation. The BLEU scores of all models are comparable (apart from the Caching Sentence architecture). Latent Grouping achieved the highest accuracy on the contrastive dataset for the context size of one, and Lateng Selecting and Single-encoder architectures for the context sizes of one and three, respectively. The results in terms of COMET (Rei et al., 2020) can be found in Appendix B.

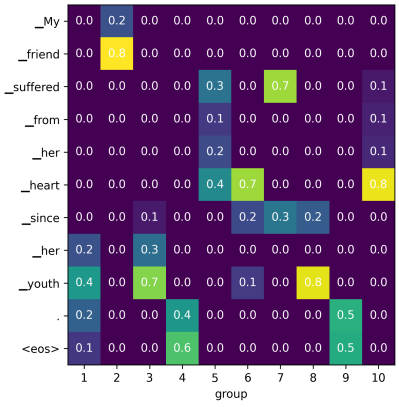
In general, Caching Tokens and Shortening models achieved higher accuracies than the Single- and Multi-encoder architectures (with the exception of the Single-encoder on English-French translation with a context size of three). Applying Latent Grouping and Latent Selecting to the cached sentence does not hurt the performance while reducing the memory footprint of the inference (Section 5.5) and increasing the interpretability of the model through the sparse assignment of tokens into groups (Section 5.4).

5.4 Token Assignment Visualization

An example visualization of groupings and selections of the Latent Grouping and Selecting architectures can be seen in Figure 3 and more can be found in Appendix C. Latent Grouping seems to group tokens according to position with nouns given a high categorization score within a group. Surprisingly, only four groups out of nine are utilized by the model. We hypothesize that the rest are used as the *no-op* tokens (Clark et al., 2019) in the context-attention when the context is not needed. Latent Selecting, by design, has to assign tokens to each group. Again, nouns seem to be included in a group more often than other parts of speech. Some



(a) Latent Grouping



(b) Latent Selecting

Figure 3: Visualization of tokens of the sentence from the ContraPro dataset grouped (3a) and selected (3b) by the model using Latent Grouping and Latent Selecting.

groups select punctuation marks and the <eos> token, which could take the role of the *no-op* tokens.

5.5 Memory Usage

We measured the memory used by the tested models as the value returned by the `torch.cuda.max_memory_allocated()` function. For clarity we omit the Caching Sentence model (as the worst performing) and the Max Pooling model (with results the same as the Avg Pooling model). Additionally, we measured the operation memory - the memory on top of the memory taken by the model during inference - on the examples from the test subset of the English-German IWSLT 2017 dataset with different numbers of context sentences. For context sizes above three, we used the models trained on the context size of three. The results are presented in Figure 4. Although the number of parameters (see Appendix A) is a dominant factor determining the overall memory usage, the operation memory grows at different paces for different architectures with the increased context size. The operational memory of the single- and multi-encoder models grows quadratically, while for caching and shortening architectures it grows linearly. Furthermore, the rate of increase is slower for shortening architectures compared to the caching architecture, which can allow the significant advantage of shortening in the setting of long sentences or large contexts.

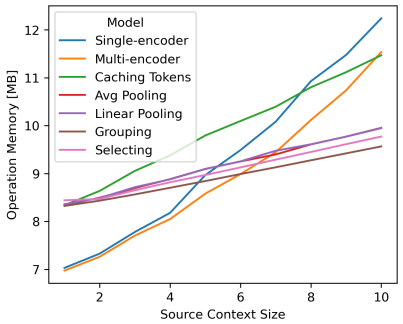


Figure 4: The mean operation memory of the models when performing inference on the examples from the English-German IWSLT 2017 test subset with the varying context sizes. For the context sizes above three, we used the models trained on the context size of three.

6 Conclusions

Caching architectures for Context-aware Machine Translation have not been widely explored in the literature so far. In this study, we show that a simple method of remembering the hidden representation of the previous sentences is comparable with more established Single- and Multi-encoder approaches in terms of BLEU and can be more effective in capturing context (up to 6 percentage points for the context size of one), measured by the accuracy on the contrastive datasets. The downside of caching methods is the diminishing returns in terms of context usage with increased context size.

Pooling-based shortening of the cached sentence maintains the comparable results to the caching architecture, while our introduced shortening methods - Latent Grouping and Selecting - show on average strong performance both in terms of BLEU and accuracy while maintaining slower growth of the memory usage during inference, and potential increased interpretability of the model through sparse assignment of tokens into groups. In future work, we will explore the integration of Sequence Shortening with the target-side context.

7 Limitations

Our investigation is limited to the source-side context. There exist linguistic phenomena that can only be addressed by using target-side context (Voita et al., 2019b). While both caching and shortening could be applied to the target side as well, we do not provide an empirical evaluation of the performance of this approach.

Furthermore, we do not apply sentence-level pre-training to our models. Architectures using Sequence Shortening could benefit from multiple stages of pre-training.

Lastly, our experiments involve language pairs from the same language family (English to German and English to French). We trained the models using the relatively low-resource datasets (IWSLT 2017) and the contrastive datasets used in this work target only the pronoun disambiguation task.

References

Ruchit Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 31–40.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3442–3455, Online. Association for Computational Linguistics.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Linqing Chen, Junhui Li, Zhengxian Gong, Min Zhang, and Guodong Zhou. 2022. One type context is not enough: Global context-aware neural machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(6).

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Zihang Dai, Guokun Lai, Yiming Yang, and Quoc Le. 2020. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *Advances in neural information processing systems*, 33:4271–4282.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yukun Feng, Feng Li, Ziang Song, Boyuan Zheng, and Philipp Koehn. 2022. Learn to remember: Transformer with recurrent memory for document-level machine translation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1409–1420, Seattle, United States. Association for Computational Linguistics.

Patrick Fernandes, Kayo Yin, Emmy Liu, André Martins, and Graham Neubig. 2023. When does translation require context? a data-driven, multilingual exploration. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 606–626, Toronto, Canada. Association for Computational Linguistics.

Patrick Fernandes, Kayo Yin, Graham Neubig, and André F. T. Martins. 2021. Measuring and increasing context usage in context-aware machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6467–6478, Online. Association for Computational Linguistics.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal,

and Ashish Verma. 2020. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR.

Christian Hardmeier. 2012. Discourse in statistical machine translation: A survey and a case study. *Discours-Revue de linguistique, psycholinguistique et informatique*, 11.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Jingjing Huo, Christian Herold, Yingbo Gao, Leonard Dahlmann, Shahram Khadivi, and Hermann Ney. 2020. Diving deep into context-aware neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 604–616, Online. Association for Computational Linguistics.

Yongkeun Hwang, Hyeongu Yun, and Kyomin Jung. 2021. Contrastive learning for context-aware neural machine translation using coreference information. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1135–1144, Online. Association for Computational Linguistics.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

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.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3512–3518, Online. Association for Computational Linguistics.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

António Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020. Document-level neural MT: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal. European Association for Machine Translation.

Lorenzo Lupo, Marco Dinarelli, and Laurent Besacier. 2022. Divide and rule: Effective pre-training for context-aware multi-encoder translation models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4557–4572, Dublin, Ireland. Association for Computational Linguistics.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3505–3511, Online. Association for Computational Linguistics.

Suvodeep Majumde, Stanislas Lauly, Maria Nadejde, Marcello Federico, and Georgiana Dinu. 2022. A baseline revisited: Pushing the limits of multi-segment models for context-aware translation. *arXiv preprint arXiv:2210.10906*.

André FT Martins and Ramón F Astudillo. 2016. From softmax to sparsemax: a sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 1614–1623.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Fabien Mathy and Jacob Feldman. 2012. What's magic about magic numbers? chunking and data compression in short-term memory. *Cognition*, 122(3):346–362.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics.

George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.

Makoto Morishita, Jun Suzuki, Tomoharu Iwata, and Masaaki Nagata. 2021. Context-aware neural machine translation with mini-batch embedding. In

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 2513–2521, Online. Association for Computational Linguistics.

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 *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Lukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. 2022. Hierarchical transformers are more efficient language models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1559–1571, Seattle, United States. Association for Computational Linguistics.

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.

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

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.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. 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.

Sandeep Subramanian, Ronan Collobert, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2020. Multi-scale transformer language models. *arXiv preprint arXiv:2005.00581*.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*.

H. S. Terrace. 2002. *The Comparative Psychology of Chunking*, pages 23–55. Springer US, Boston, MA.

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raul Vazquez, and Sami Virpioja. 2022. Democratizing machine translation with opus-mt. *arXiv preprint arXiv:2212.01936*.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*, 5:87–99.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine 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 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Billy T. M. Wong and Chunyu Kit. 2012. Extending machine translation evaluation metrics with lexical cohesion to document level. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1060–1068, Jeju Island, Korea. Association for Computational Linguistics.

Xueqing Wu, Yingce Xia, Jinhua Zhu, Lijun Wu, Shufang Xie, and Tao Qin. 2022. A study of bert for context-aware neural machine translation. *Machine Learning*, 111(3):917–935.

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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 788–801, Online. Association for Computational Linguistics.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1081–1087, Online. Association for Computational Linguistics.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2021. Towards making the most of context in neural machine translation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3983–3989.

A Models and Training Details

To implement and train our models we used fairseq framework (Ott et al., 2019) and based our code on the codebase of Fernandes et al. (2021). All models were based on the transformer-base configuration. The shared hyper-parameters are presented in Table 4. We trained each model on a single GPU (NVIDIA GeForce RTX 3090 24GB).

For Latent Grouping and Shortening, we used a categorizing FFN with 512 hidden units, the number of inputs equal to the Embed Dim, and the number of outputs equal to the number of groups. Table 5 shows the number of parameters for each model.

Hyper-parameter	Value
Encoder Layers	6
Decoder Layers	6
Attention Heads	8
Embed Dim	512
FFN Embed Dim	2048
Dropout	0.3
Share Decoder In/Out Embed	True
Optimizer	Adam
Adam Betas	0.9, 0.98
Adam Epsilon	1e-8
Learning Rate	5e-4
LR Scheduler	Inverse Sqrt
LR Warmup Updates	2500
Weight Decay	0.0001
Label Smoothing	0.1
Clip Norm	0.1
Batch Max Tokens	4096
Update Frequency	8
Max Epoch	-
Patience	5
Beam	5
Max Vocab Size	20000
Seed	42

Table 4: The shared hyper-parameters of the tested models.

Model	Parameters
Sentence-level	64.42M
Single-encoder	64.42M
Multi-encoder	83.33M
Caching Tokens	71.25M
Caching Sentence	71.26M
Shortening - Max Pooling	72.83M
Shortening - Avg Pooling	72.83M
Shortening - Linear Pooling	73.35M
Shortening - Grouping	72.58M
Shortening - Selecting	72.58M

Table 5: The number of parameters in the tested models.

B Extended Results

Apart from BLEU and contrastive dataset accuracy presented in Section 5, we also measured COMET (Rei et al., 2020) based on `Unbabel/wmt22-comet-da` model (Rei et al.,

2022). See Tables 6 and 7 for the results on En-De and En-Fr respectively.

C Groupings and Selections Visualization

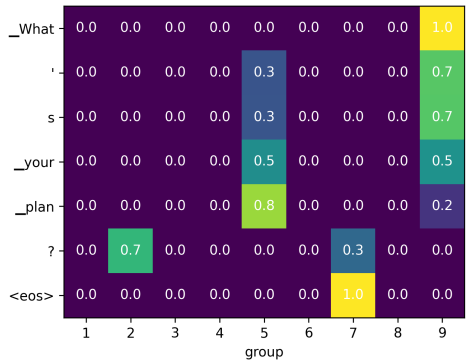
The visualizations of groupings and selections done by the models using Latent Grouping and Selecting of the additional examples from the ContraPro dataset (Müller et al., 2018) can be found in Figure 5. Figure 6 shows the visualizations of the groupings and selections of the sentences from the contrastive dataset by Lopes et al. (2020).

Model	**Context: 0**		
Sentence-level	0.7778		
Model	**Context: 1**	**Context: 2**	**Context: 3**
Single-encoder	0.7831	0.7789	0.7758
Multi-encoder	0.7831	**0.7871**	**0.7856**
Caching Tokens	0.7806	0.7776	0.7821
Caching Sentence	0.7712	0.7640	0.7673
Shortening - Max Pooling	0.7743	0.7772	0.7799
Shortening - Avg Pooling	0.7774	0.7770	0.7844
Shortening - Linear Pooling	0.7757	0.7745	0.7823
Shortening - Grouping	**0.7842**	0.7828	0.7811
Shortening - Selecting	0.7774	0.7826	0.7836

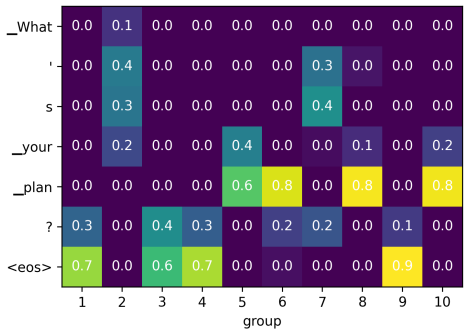
Table 6: Results in terms of COMET (Rei et al., 2020) based on `Unbabel/wmt22-comet-da` model (Rei et al., 2022) of the **En-De** IWSLT 2017 experiment.

Model	**Context: 0**		
Sentence-level	0.7943		
Model	**Context: 1**	**Context: 2**	**Context: 3**
Single-encoder	0.7930	**0.7979**	0.7913
Multi-encoder	**0.7968**	0.7934	0.7934
Caching Tokens	0.7923	0.7935	0.7945
Caching Sentence	0.7845	0.7654	0.7737
Shortening - Max Pooling	0.7911	0.7913	**0.7974**
Shortening - Avg Pooling	0.7920	0.7924	0.7952
Shortening - Linear Pooling	0.7933	0.7951	0.7927
Shortening - Grouping	0.7933	0.7976	0.7921
Shortening - Selecting	0.7951	0.7945	0.7935

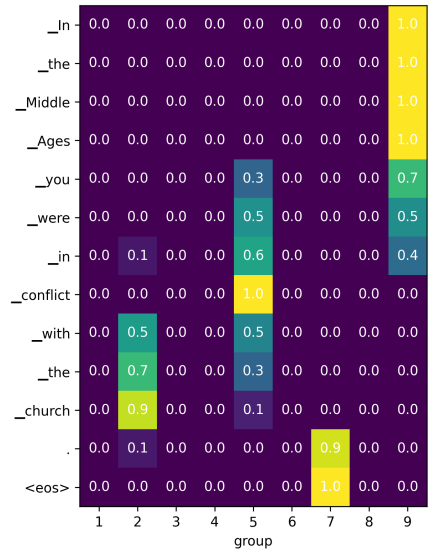
Table 7: Results in terms of COMET (Rei et al., 2020) based on `Unbabel/wmt22-comet-da` model (Rei et al., 2022) of the **En-Fr** IWSLT 2017 experiment.



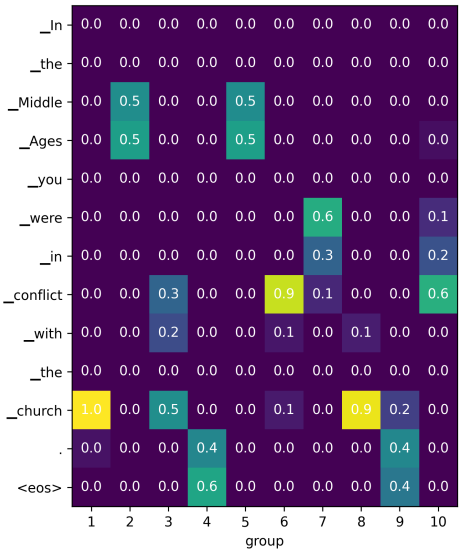
(a) Latent Grouping



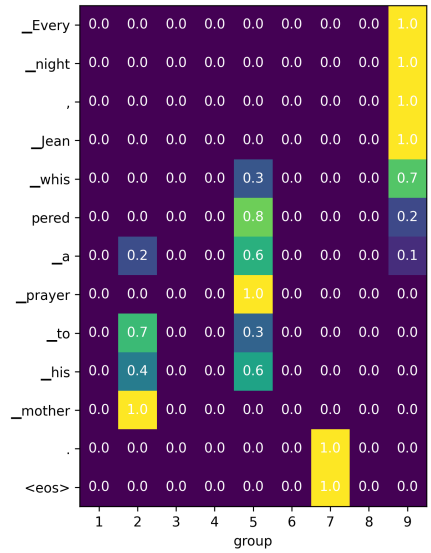
(b) Latent Selecting



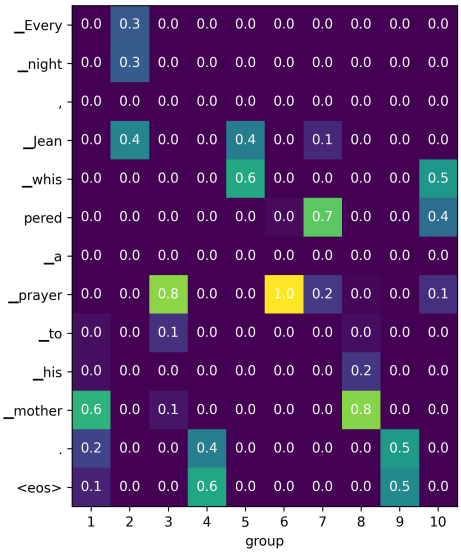
(c) Latent Grouping



(d) Latent Selecting

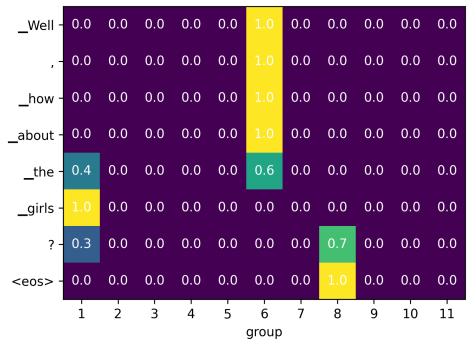


(e) Latent Grouping

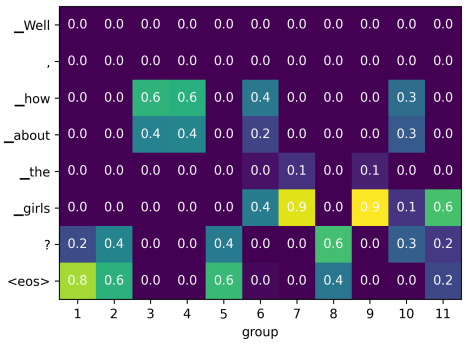


(f) Latent Selecting

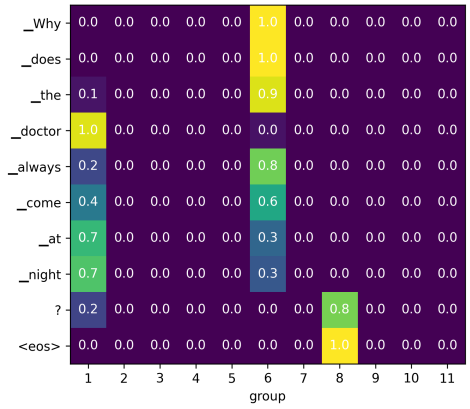
Figure 5: Visualization of tokens of the sentences from the ContraPro dataset (Müller et al., 2018) grouped (5a, 5c, 5e) and selected (5b, 5d, 5f) by the model using Latent Grouping and Latent Selecting.



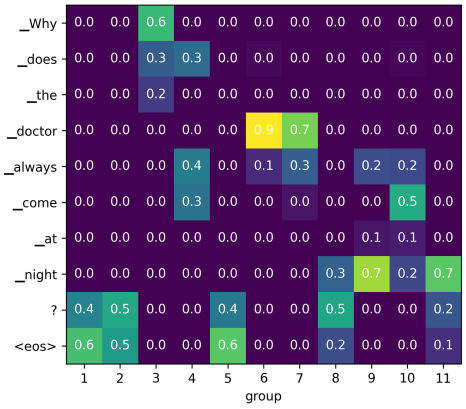
(a) Latent Grouping



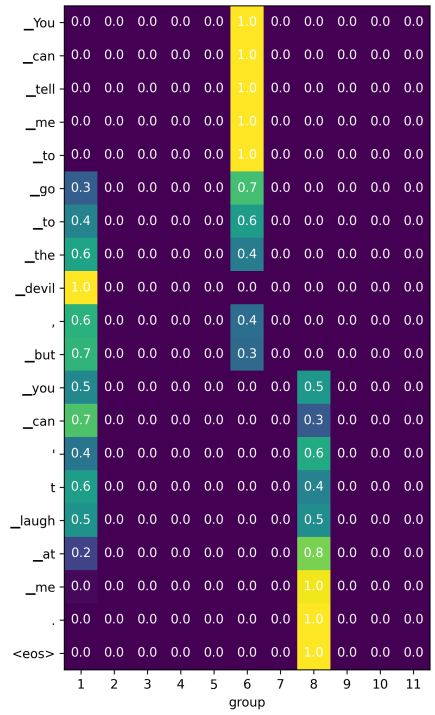
(b) Latent Selecting



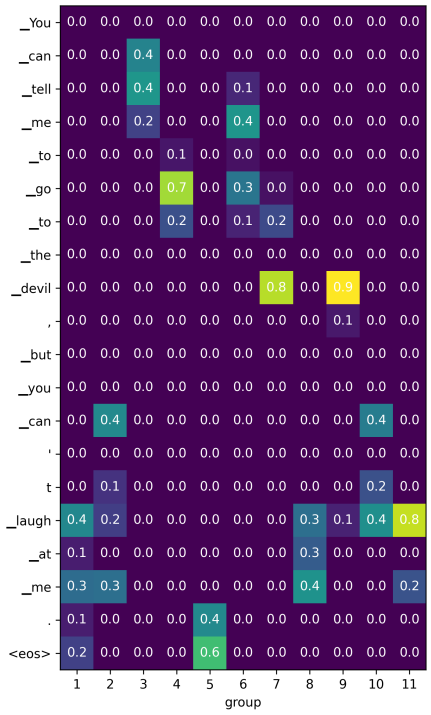
(c) Latent Grouping



(d) Latent Selecting



(e) Latent Grouping



(f) Latent Selecting

Figure 6: Visualization of tokens of the sentences from the contrastive dataset by Lopes et al. (2020) grouped (6a, 6c, 6e) and selected (6b, 6d, 6f) by the model using Latent Grouping and Latent Selecting.