Machine Translation with Unsupervised Length-Constraints

Jan Niehues jan.niehues@maastrichtuniversity.nl
Department of Data Science and Knowledge Engineering (DKE), Maastricht University, Maastricht, The Netherlands

Abstract

We have seen significant improvements in machine translation due to the usage of deep learning. While the improvements in translation quality are impressive, the encoder-decoder architecture enables many more possibilities. In this paper, we explore one of these, the generation of constrained translation. We focus on length constraints, which are essential if the translation should be displayed in a given format.

In this work, we propose an end-to-end approach for this task. Compared to a traditional method that first translates and then performs sentence compression, the text compression is learned completely unsupervised. We address the challenge of data availability as well as investigate several methods to integrate the constraints into the model. By combining the idea with zero-shot multilingual machine translation, we are also able to perform unsupervised monolingual sentence compression.

Using the proposed approach, we are able to improve the translation quality for translation with length constraints as well as for monolingual length compression. In addition, the results are confirmed by a human evaluation.

1 Introduction

Neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2014) exploits neural networks to directly learn to transform sentences in a source language to sentences in a target language. This technique has significantly improved the quality of machine translation (Bojar et al., 2016; Cettolo et al., 2015). The advances in quality also allow for the application of this technology to new real-world applications.

While research systems tend to purely focus on a high translation quality, real-world applications often have additional requirements for the output of the system. One example is the mapping of markup information from the source text to the target text (Zenkel et al., 2019). In this work, we will focus on another use case, the generation of translations with given length constraints. Thereby, we focus on compression. That means the target length is shorter than the actual length of the translation. When translating from one language to another, the length of the source text is usually different from the length of the target text. While for most applications of machine translation this does not pose a problem, for some applications this significantly deteriorates the user experience. For example, if the translation should be displayed in the same layout as the source text (e.g. in a website), it is advantageous if the length stays the same. Another use case are captions for videos. A human is only capable of reading text up to a certain speed. For an optimal user experience, it is therefore not only important to present an accurate translation, but also to present the translation with a maximum number of words.

A first approach to address this challenge would be to use a cascade of a machine translation and sentence compression system. In this case, we would need training data to train the machine translation system and additional training data to train the sentence compression system. It is very difficult and sometimes even impossible to collect the training data for the sentence compression task. Furthermore, we need a sentence compression model with a parametric length reduction ratio. For a supervised model, we would therefore need examples with different length reduction ratios. Therefore, this work focuses on unsupervised sentence compression. Compared to related work, in this method we even do not assume to have any compressed sentences. So we need to learn how to compress sentences without having seen any compressed sentence in training.

While our work focuses on the end-to-end approach to translation combined with sentence compression, monolingual sentence compression is another important task. For example, human-generated captions are often not an accurate transcription of the audio, but in addition the text is shortened. This is due to cognitive processing constraints. The user is able to listen to more words in a given time than he or she can read in the same amount of time. When combining the length-constrained machine translation with the idea of zero-shot machine translation, the proposed method is also able to perform monolingual sentence compression. In addition, by adjusting the loss function we are able to use the same framework to perform text simplification.

The main contribution of this work is an end-to-end approach to length-constrained translation by jointly performing machine translation and sentence compression. We are able to show that for this task an end-to-end approach outperforms the cascade of machine translation and unsupervised sentence compression.

Therefore, two contributions are essential. First, by using pseudo-supervised training on standard parallel data, we are able to learn to compress without ever showing the model a compress sentence. This is achieved by making the model aware of properties (here the maximum length) that the output must fulfil. The second contribution is the adaptation of the architecture that the model is able to fulfil the properties also there is a mismatch between its influence in training and in testing. While it is straightforward to fulfil it during training, it can be difficult during decoding.

A third contribution of this work is to extend the presented approach to unsupervised monolingual sentence compression. By combining the presented approach with multilingual machine translation, we are able to also generate paraphrases with a given length constraint. The investigation shows that a system that is trained on several languages is able to successfully generate monolingual paraphrases.

2 Constrained decoding

In the targeted scenario, there is no available training data with the constraints of interest. Therefore, we need to teach the system something about the output that it cannot directly learn from the data. We investigate different methods that enable the model to fulfil the constraints without learning them from the data.

The main application is length-constrained translation. That means that we want to generate a translation with a given target length. Therefore we focus on the case of shortening the translations. While the length can be measured in words, subword tokens or letters, in the experiments we measured the length by subword tokens.

A straightforward approach is to disregard the constraints during training and search for the most probable translation respecting the constraints during decoding. We do this by restricting the search space to generate only translations with a given length. The length of the output is modeled by the probability of the end-of-sentence (EOS) token. By modifying this probability, we introduce a hard constraint that is always fulfilled.

Afterwards, we propose the length-aware model. This uses two techniques that are able to learn how to fulfil the constraints without ever seeing a sentence that was generated with the constraints. This enables us to train the model on standard parallel data.

We use pseudo-supervised training and assign the matching constraints to existing parallel data. Furthermore, by adapting the architecture, we address the train-test mismatch between the constraints in training and in decoding.

It is worth noting that the length-aware model uses the constraints as soft constraints, where translations fulfilling the constraints are preferred but other ones could also be generated. In contrast, they are modeled as hard constraints when adapting the search. In this case, only translation that fulfill the constraints are generated. Therefore, both methods can also be combined.

2.1 Restricted search space

A first strategy to incorporate the additional length constraints is to ignore them during training and restrict the inference-time search space to hypotheses that fulfill the constraint. For length constraints, this can be achieved by manipulating the end-of-sentence token probability. First, we need to ensure that the EOS token is not generated before the desired length of output J. This can be ensured by setting the probability for the end-of-sentence token to zero for all positions before the desired length and re-normalizing the probability.

$$p'(y_j|x_1,\ldots,x_I,y_1,\ldots,y_{j-1}) = \begin{cases} \frac{p(y_j|x_1,\ldots,x_I,y_1,\ldots,y_{j-1})}{1-p(EOS|x_1,\ldots,x_I,y_1,\ldots,y_{j-1})} & y_j \neq EOS \\ 0 & y_j = EOS \end{cases} \quad (1)$$

Finally, we ensure to stop the search at the desired length by setting the probability of the end-of-sentence token to one if the output sequence has reached this length.

$$p'(y_j|x_1,\ldots,x_I,y_1,\ldots,y_{j-1}) = \begin{cases} 0 & y_j \neq EOS \\ 1 & y_j = EOS \end{cases} \quad (2)$$

While this approach will guarantee that the output of the translation systems always meets the length condition (hard constraint), it also has one major drawback. Until the system reaches the constrained length, the system is not aware of how many words it is still allowed to generate. Therefore, it is not able to shorten the beginning of the sentence in order to fulfil the length constraint.

Motivated by this observation, we investigate methods to integrate the length constraint into the model and not only apply it during inference.

2.2 Length-aware model

The main idea of the length-aware model is that it should be aware of the output length through the whole decoding process. Therefore, the model needs as input in addition to the source sentence $X = x_1,\ldots,x_I$ the desired target length J. Then the model can decide what to generate based on the source text as well as only the available space given by the length constraint. However, this poses the challenge that we also need to train the model using data with constraints. These challenges are addressed by using pseudo-supervised training, where the constraints are added to existing parallel data and by integrating the length into the model. We will investigate methods to encode the length globally for the full sentence as well as methods to encode the remaining length locally at each decoding step.

2.2.1 Pseudo-supervised training

In contrast to only restricting the search space, for the length-aware model we also need the target length during training. Therefore, the first challenge we need to address when including the length constraints into the model itself is the question of training data. While there is large amounts of parallel training data, it is hard to acquire training data with length constraints. Therefore, we investigate methods to train the model with standard parallel training data. In contrast to other unsupervised methods, we are missing not only parallel data between the input and the output, but we have no data for the output. So we need to learn how to compress a sentence without ever seeing a compressed sentence.

Motivated by the success of adding the length in the use case where the output length should not be shortened but similar to the input length (Lakew et al., 2019), we perform the training by a type of pseudo-supervision. For each source sentence, in training, we also know the translation and therefore its length. The main idea is that we now assume this sentence was generated with the constraint to generate a translation with exactly the length of the given translation. Of course, this is mostly not the case. The human translator generated a translation that appropriately expresses the meaning of the source sentence and not a sentence that fulfills the length constraints.

Therefore, we have a mismatch between training and testing conditions and the learning is more difficult. While during training the given length can relatively easily be predicted by the expected length when expressing all source content in the target language, this is no longer true for testing. Due to the condition in training the system might learn to simply ignore the length information and instead generate a normal translation putting all the information of the source sentence into the target sentence. In this case, we would not have the possibility to control the target length by specifying our desired length.

2.2.2 Length representation

To address this problem, we investigate three different methods to represent the target length in the model. The motivation is thereby to ensure that the model uses the additional length information although it might not strictly necessary during training. Thereby, the training examples consist of a source sentence $X = x_1, \ldots, x_I$, a target sentence $Y = y_1, \ldots, y_J$ and the target length J.

Source embedding A first method is to model the target length globally for the whole sentence. This can be achieved by including the target length into the source sentence as an additional token. This is motivated by successful approaches for multilingual machine translation (Ha et al., 2016), domain adaptation (Kobus et al., 2017) and formality levels (Sennrich et al., 2016a). We change the training procedure to not use X as the input to the encoder of the NMT system, but instead J, X. In this way, the encoder will learn an embedding for each target length seen during training.

There are two challenges using this approach. First, the dependency between the described length J and the output Y is quite long within the model. Therefore, the model might ignore the information and just learn to generate the best translation for a given source sentence. Secondly, the representations for all possible target lengths are independent from each other. This poses a special challenge for long sentences which occur less frequently, e.g. there will be less sentence with length 63 than with length 9 and therefore the embedding of these lengths will not be learned as well as the frequent ones.

Target embedding We address the first challenge by integrating the length constraint directly into the decoder. In this case we model locally at each decoding step by encoding the number of words remaining to be generated. This is motivated by similar approaches to supervised sentence compression (Kikuchi et al., 2016) and zero-shot machine translation (Ha et al., 2017).

We incorporate the information of the number of remaining target words at each target position. For one, this should ensure that the length information is not lost during the decoding process. Secondly, by embedding smaller numbers which occur more frequently in the corpus towards the end of the sentence, the problem of rare sentence lengths does not matter that much.

Formally, at each decoder step j the baseline model starts with the word embedding of the last target word y_{j-1}. In the original transformer architecture (Vaswani et al., 2017), the positional encoding is applied on top of the embedding to generate the first hidden representation

$$h_0 = pos(emb(y_{j-1}), j). \tag{3}$$

In our proposed architecture, we include the number of remaining target words to be generated $J - j$. We concatenate h_0 with the length embedding and then apply a linear translation and a non-linearity to reduce the hidden size to the one of the original word embedding

$$h'_0 = relu(lin(cat(h_0, lenEmb(J - j))). \tag{4}$$

The proposed architecture allows the model to consider the number of remaining target words at each decoding step. While the baseline model will only cut the end of the sentences, the model is able to shorten already at the beginning of the sentence.

Positional encoding Finally, we also address the challenge of representing sentence lengths that are less frequent. The transformer architecture introduced the positional encoding. This encodes the position within the sentence using a set of trigonometric functions. While their method encodes the position relative to the start of the sentence, we follow Takase and Okazaki (2019) to encode the position relative to the end of the sentence. Thereby, at each position we encode the number of remaining words of the sentence. Formally, we replace $h_0 = pos(emb(y_{j-1}), j)$ by $h_0^* = pos(emb(y_{j-1}), J - j)$.

2.3 Additional constraints

Besides constraining the number of words, other constraints can be implemented as easily using the same framework. In this work, we show this by limiting the number of complex and difficult words. One use case is the generation of paraphrases in simplified language. A metric to measure text difficulty, the Dale-Chall Readability metric (Chall and Dale, 1995), for example, counts such difficult words. In an NMT system, longer words are typically split into subword units by Byte Pair Encoding (BPE) (Sennrich et al., 2016b). A complex word like *marshmallow* is split into several parts, for instance *mar@@ shm@@ allow*, where @@ indicates that the word is not yet finished.

The idea to generate simpler text is now to limit the sub-word tokens that do not end a word (the ones ending on @@). This can be implemented by only counting the words that end on @@. If the target sentence would for example by *I like mar@@ shm@@ allow*, the target count would be 2.

When encoding the remaining length in the decoder, we would now not reduce it by 1 at each step, but only if the token is ending on @@. So the length sequence used in decoding would be 2 2 2 1 0.

During inference, we would now always try to generate sequences without splitted words by inputting the target length of 0. Since it is a soft constraint, the model can still generate subwords, if the other model strongly suggests that.

As for the length constrained decoding, we also would perform pseudo-supervised training and thereby be able to train our model on the default parallel data.

Reference:	It might sound like it's a bad thing.
Baseline:	But it might sound like
Constrained:	It sounds really bad .

Table 1: Example of constrained translation

3 Evaluation

The lack of suitable data is not only a challenge for training but also for evaluation. The default approach to evaluating a machine translation system is to compare the output of the system with human translation using some automatic metric, e.g. BLEU (Papineni et al., 2002).

In our case, we would need to have a human-generated translation, which also fulfills the additional constraints. For example, translation with a length that is shortened to 80% of the input. Since this type of translation data is not available, we investigate methods to compare the length-constrained output of the system with standard human translation that do not fulfill any specific constraints.

3.1 Word matching metrics

While there is a significant amount of research in automatic metrics for machine translation (Ma et al., 2018, 2019), BLEU is still the most commonly used metric. Therefore, a first approach would be to use BLEU to compare the automatic translation with length constraints with the human translation without constraints. If we were using length constraints, this would lead to low BLEU scores due to the length penalty of the metric. But since all systems must fulfill the length constraint, the penalty would be the same for all output and we could still compare between the different outputs.

A problem of using BLEU scores as evaluation metrics in this task is illustrated by the example translations in Table 1. The baseline system only uses the length constraint for restricting the search space. In the constrained system, we are using the length constraint also as additional embeddings in the decoder. Looking at this example sentence, a human would rate the constrained translation better than the baseline translation. The problem of the latter model is that it often generates a prefix of a full translation. While this does not lead to a good constrained translation, it still leads to a relatively high BLEU score. In this case, we have one matching 4-gram, 2 tri-gram, 3 bigrams and four unigrams.

In contrast, the length-constrained model only contains words matching the reference scattered over the sentence. Therefore, in this case, we only have two unigram matches. Guided by this observation, we used different metrics to evaluate the models.

3.2 Embedding-based metrics

In order to address the challenges mentioned in the last subsection, we used metrics that are based on sentence embeddings instead of word or character-based representation of the sentence. This way it is no longer important that the words occur in the same sequence in automatic translation and reference. Based on the performance of the automatic metrics in the WMT Evaluation campaign in 2018, we used RUSE (Shimanaka et al., 2019) metric. It uses sentence embeddings from three different models: InferSent, Quick-Thought and Universal Sentence Encoder. Then the quality is estimated by an MLP based on the representation of the hypothesis and the reference translation. The hyper parameters (number of layers, hidden size, batch size, dropout rate) were optimized on the development set of the WMT Evaluation campaign and the MLP was not retrained for this task.

4 Experiments

4.1 Data

We train our systems on the TED data from the IWSLT 2017 multilingual evaluation campaign (Cettolo et al., 2017). The data contains parallel data between German, English, Italian, Dutch and Romanian. We create three different systems. The first system is only trained on the German-English data, the second one is trained on German-English and English-German data and the last one is trained on {German, Dutch, Italian, Romanian} and English data in the both directions.

The data is preprocessed using standard MT procedures including tokenization, truecasing and BPE with 40K codes. For model selection, the checkpoints performing best on the validation data (dev2010 and tst2010 combined) are averaged, which is then used to translate the tst2017 test set.

In the experiments, we address two different targeted lengths. Thereby the length of a sentence is measured by the number of subword tokens. In order to not use any information from the reference, we measure length limits relative to the source sentence length. We aim to shorten the translation to produce output that is 80% and 50% of the source sentence length. While in the first case, most information can still be conveyed, we wanted to see if the model is able to concentrate really on the important parts when shorting by half the length.

4.2 System

We use the standard transformer architecture (Vaswani et al., 2017) and increase the number of layers to eight. The layer size is 512 and the inner size is 2048. Furthermore, we apply word dropout (Gal and Ghahramani, 2016) with $p = 0.1$. In addition, layer dropout is used with $p = 0.2$ as in Pham et al. (2019). We use the same learning rate schedule as in the original work. The implementation is available on github[1]. All systems were always trained from scratch with random initialization.

4.3 Task difficulty

In an initial set of experiments, we assess the difficulty of having the additional length constraints. Therefore, we used the length of the human reference translation as a first target length. One could even argue that should make the typical machine translation easier, since some information about the translation is known. The results of this experiment are shown in Table 2. Since we do not perform compression in this experiment, the aforementioned problem with BLEU should not apply here.

Model	BLEU	RUSE
Baseline	30.80	−0.085
Only Search	28.32	−0.124
Source Emb	28.56	−0.126
Decoder Emb	27.88	−0.140
Decoder Pos	28.80	−0.138

Table 2: Using oracle length

However, the results indicate that the baseline system achieves the best BLEU score as well as the best RUSE score. All other models generate translations that perfectly fit the desired target length, but this leads to a drop in translation quality. Therefore, even if the target length

[1] https://github.com/jniehues-kit/NMTGMinor/tree/DbMajor

is the same as the one of the reference translation, the restriction increases the difficulty of the problem. One reason could be that the machine translation system rarely generates translations which exactly match the reference. By forcing the translation to have an exact predefined length, we are increasing the difficulty of the problem.

4.4 Length representation

In a first series of experiments for length constrained output, we analyzed the different techniques to encode the length of the output. First, we are interested in whether the different length representations are able to enforce an output that has the length we are aiming at (soft constraints). For the German to English translation task, the length of the different encoding versions are shown in Table 3. We define the length as the average difference between the targeted output given in BPE units and the output of the translation system.

First, without adding any constraints, the models generate translations that differ by 3.9 and 10.29 words from the targeted length. By specifying the length in the source side, we can reduce the length difference to half a word in the case of a targeted length of 80% and one and a half words in the case of 50% of the source length. The models using the decoder embeddings and the decoder positional encoding were able to nearly perfectly generate translation with the correct number of words.

Encoding	Avg. length difference	
	80%	50%
Baseline	3.90	10.29
Source Emb	0.55	1.40
Decoder Emb	0.07	0.16
Decoder Pos	0.09	0.19

Table 3: Avg. Length distance

Besides fulfilling the length constraints, the translations need to be accurate. Since we wanted to have a fair comparison, we evaluated the output when using a restricted search space, so that only translations with the correct number of words are generated (hard constraints). The results are summarized in Table 4

Encoding	RUSE	
	80%	50%
Baseline	−0.272	−0.605
Source Emb	−0.263	−0.587
Decoder Emb	−0.2469	−0.555
Decoder Pos	−0.2598	−0.577

Table 4: German-English translation quality

As shown in the results, we see improvements in translation quality when using the source embedding within the encoder. We have further improvements if we represent the targeted length within the decoder. In this case, we can improve the RUSE score by 2% and 5% absolute. The decoder encodings perform similarly, with small advantage for using embeddings and not positional encodings.Therefore, in the remaining of the experiments we use the embeddings.

4.5 Multi-lingual

In a second series of experiments, we combine the constrained translation approach with multi-lingual machine translation. The combination of both offers the unique opportunity to perform unsupervised sentence compression. We can treat the translation of English to English as a zero-shot direction (Johnson et al., 2017; Ha et al., 2016). This has not been addressed in traditional multi-lingual machine translation, since in this case the model will often just copy the source sentence to the target one. By adding the length constraints, we force the mode to reformulate the sentence in order to fulfil the length constraint.

The results for these experiments are shown in Table 5. In this case, we compared three scenarios. First, a model trained only on translations from German to English. Secondly, a model trained to translate from German to English and English to German. Finally, a model trained on four languages to and from English.

	Target Length 0.8				Target Length 0.5			
	Baseline		Dec. Emb		Baseline		Dec. Emb	
Model	DE-EN	EN-EN	DE-EN	EN-EN	DE-EN	EN-EN	DE-EN	EN-EN
DE-EN	−0.272		−0.247		−0.587		−0.554	
DE+EN	−0.264	−0.817	−0.223	−0.905	−0.598	−0.954	−0.523	−0.978
All	−0.225	−0.102	−0.214	0.020	−0.560	−0.525	−0.548	−0.481

Table 5: Multi-lingual systems

First of all, since the models are trained on relatively small data, we always gain when using more language pairs. Secondly, for all models training from German to English, the decoder embedding is clearly better than the baseline. Finally, to perform paraphrasing, we need a multilingual system with several language pairs. Both models trained only on the German to English and English to German data fail to generate adequate translation. In contrast, if we look at the translation from English to English for the multilingual model, the scores are clearly better than the ones from German to English. Furthermore, again, the system with decoder embeddings is clearly better than the baseline system.

In addition, we performed the same experiment with a target length of half the source length (Table 5). Although the absolute scores are significant lower since the model has to reduce the length further, the tendency is the same for this direction.

4.6 End2End vs. Cascaded

Length	Model	DE-EN	EN-EN
0.8	End2End	−0.247	0.020
	Cascade	−0.259	−0.118
	Cascade Fix. Pivot		−0.166
0.5	End2End	−0.555	−0.481
	Cascade	−0.575	−0.521
	Cascade Fix. Pivot		−0.544

Table 6: Comparison of End-to-End and Cascaded approach

In this work, we are able to combine machine translation and sentence compression. In a third series of experiments, we wanted to investigate the advantage of modelling it in an end-to-end fashion compared to a cascade of different models. We performed this investigation again

for two tasks: German to English and English to English.

The cascade system for German to English, first translates the German text to English with a baseline machine translation system. In a second step, the output is compressed with the multi-lingual MT system. For the English-to-English system, the cascade system removes the zero-shot condition. Therefore, we first translate from English to German with the baseline system and then translate with length contrasted from German to English. In *cascade fix pivot* also the English to German system already fulfills the length constraint.

As shown in Table 6, in all conditions, the end-to-end approach outperforms the cascaded version. This is especially the case for the English-to-English machine translation. Compared to multi-lingual machine translation, for these tasks it seems to be beneficial to perform the zero-shot tasks instead of using a pivot language.

4.7 Simplification

Metric	DE-EN		DE+EN		All	
	Base	Simp.	Base	Simp.	Base	Simp
BPE tokens	1961	1053	1978	1041	1899	991
DCR	7.63	7.47	7.69	7.5	7.66	7.45
FRE	83.86	86.18	84.31	85.49	82.98	85.59
BLEU	30.80	30.62	32.25	31.38	32.84	31.29
RUSE	−0.085	−0.092	−0.082	−0.080	−0.042	−0.084

Table 7: Simplification

In the last series of experiments (Table 7), we investigate the ability of our method to generate simpler sentences. As described in Section 2.3, we used the proposed framework to reduce the number of rare and complex words. Again, we are using the decoder embedding to represent the amount of BPE units in the sentences. We use a system for 1 language pair, 2 language pairs and the system using 8 language pairs. First, the system is able to reduce the number of BPE tokens in the text significantly. The amount of tokens is reduced by up to 48%. Since the number of tokens is nearly kept the same, this is also reflected in a better readability. This highlights also the importance of having soft constraints. In this use case, we cannot generate reasonable translations without using rare words that get split into separate subword units. However, the proposed framework is able to reduce the amount of these words.

We measure the readability using the Dale-Chall readability formula (DCR) (Chall and Dale, 1995) and the Flesch Reading Ease (FRE) (Flesch, 1948) .[2] As shown in the table, both scores indicate that the readability is increased by the proposed method. On the other hand, we see that the translation quality is only affected slightly.

4.8 Human Evaluation

In addition to the automatic evaluation, a human evaluation of the output was performed. This evaluation was performed using the multilingual machine translation system translating from German to English. Thereby, a length constraint of 80% of the source sentence was used.

Two evaluators were asked to assign a score between 0 and 100 to the translations of 15 sentences each. In Table 8 we summarized the results. We calculate the average score of both systems as well as how often the system from one model was evaluated better than the other. First, the proposed methods were evaluated better by around 5%. Secondly, when comparing

[2]The scores were calculated by the tool https://github.com/mmautner/readability

the scores for the individual sentences, the proposed method generated better translations nearly three times as often as the baseline model did.

Evaluation	Baseline	Decoder Emb
Score	68%	73%
Wins	6	17

Table 8: Human Evaluation

4.9 Qualitative Results

For the length restricted system, we also present examples in Table 9. The translations were generated with the multi-lingual system using restricted search space with 0.8 times and 0.5 times the length of the source length. The length is thereby measured using the number of subword tokens.

Source:	Und, obwohl es wirklich einfach scheint, ist es tatsächlich richtig schwer, weil es Leute drängt sehr schnell zusammenzuarbeiten.
Reference:	And, though it seems really simple, it's actually pretty hard because it forces people to collaborate very quickly.
Base 0.8:	and even though it really seems simple , it is actually really hard , because it really pushes
Dec. Emb. 0.8 :	and although it really seems simple , it is really hard because it drives people to work together .
Base 0.5 :	and even though it really seems simple , it is really hard
Dec. Emb. 0. 5:	it is really hard because it drives people to work together .
Source:	Konstrukteure erkennen diese Art der Zusammenarbeit als Kern eines iterativen Vorgangs.
Reference:	Designers recognize this type of collaboration as the essence of the iterative process.
Base 0.8:	now , traditional constructors recognize this kind of collaboration as the core
Dec. Emb. 0.8	designers recognize this kind of collaboration as the core of iterative .
Base 0.5:	now , traditional constructors recognize this kind
Dec. Emb: 0.5	developers recognize this kind of collaboration .

Table 9: Examples

In the examples we see clearly the problem of the baseline model when using a restricted search space. The model mainly outputs the prefix of the long translation and does not try to put the main content into the shorter segment. In contrast, the system using the decoder embeddings is aware when generating a word how much space it still has to fill the content. Therefore, it does not just cut part of the sentence, but compress the sentence and extract the most important part of the sentence. While the first example is more concentrating on the second part of the original sentence, the second one is focusing at the beginning. Although the model reducing the length by 50% has to remove some content of the original sentence, the sentence is still understandable.

5 Related Work

The most common approach to model the target length within NMT is the use of coverage models (Tu et al., 2016). More recently, (Lakew et al., 2019) used similar techniques to generate

translation with the same length as the source sentence. Compared to these works, we tried to reduce the length of the sentence by a larger margin and thereby have the situation where the training and testing conditions differ more. Furthermore, the use of multi-lingual machine translation allows also the generation of compressed sentences in the same language. This work on length-controlled machine translation is strongly related to sentence compression, where the compression is performed in the monolingual case. First approach used rule-based approaches (Dorr et al., 2003) for extractive sentence compression. In abstractive compression methods using syntactic translation (Cohn and Lapata, 2008) and phrase-based machine translation were investigated (Wubben et al., 2012). The success of encoder-decoder models in many areas of natural language processing (Sutskever et al., 2014; Bahdanau et al., 2014) motivated their successful application to sentence compression. (Kikuchi et al., 2016) and (Takase and Okazaki, 2019) investigated an approach to directly control the output length. Although their methods use similar techniques to ours, the model is trained in a supervised way. Motivated by recent success in unsupervised machine translation (Artetxe et al., 2018; Lample et al., 2018), a first approach to learn text compression in an unsupervised fashion was presented in Fevry and Phang (2018). Text compression in a supervised fashion for subtitles was investigated in Angerbauer et al. (2019).

In contrast to text compression, the combination of readability and machine translation has been researched recently. (Agrawal and Carpuat, 2019) presented an approach to model the readability using source side annotation. In contrast to our work, they concentrated on the scenario where manually created training data is available. In Marchisio et al. (2019) the authors specified the desired readability difficulty either by a source token or through the architecture by different encoders. While they concentrate on a single task and have only a limited number of difficulty classes, the work presented here is able to handle a huge number of possible output classes (e.g. in text compression the number of words) and can be applied for different tasks.

6 Conclusion

In this work, we investigated the challenge of generating translation with additional constraints. The main difficulty we addressed is the availability of training data. It is not only hard to acquire parallel data with target sides constraints, but even monolingual data which was generated respecting additional constraints is rarely available.

We address this problem by using pseudo-supervised training on standard parallel data. Instead of generating the translation with constraints, we set the constraints in a way that they are fulfilled by the existing translation. Thereby, we are able to learn to generate compressed sentences without ever seeing compressed sentences in training.

The approach results in a mismatch between training and test conditions. While in training the translation can also be correctly generated by ignoring the constraints, this is no longer the case in testing. We address this issue by adapting the architecture of the sequence-to-sequence model. A detailed evaluation using automatic and human evaluation shows the success of the presented approach.

Finally, we show the possibility to extend the presented approach to related tasks. In combination with zero-shot multi-lingual machine translation, the method is also able to perform monolingual sentence compression. Furthermore, by varying the cost function, we are able to also address other tasks like text simplification.

References

Agrawal, S. and Carpuat, M. (2019). Controlling Text Complexity in Neural Machine Translation. *arXiv:1911.00835 [cs]*. arXiv: 1911.00835.

Angerbauer, K., Adel, H., and Vu, T. (2019). Automatic Compression of Subtitles with Neural Networks and its Effect on User Experience. In *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, pages 594–598.

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2018). Unsupervised Neural Machine Translation. In *International Conference on Learning Representations*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Névéol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. (2016). Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Cettolo, M., Federico, M., Bentivoldi, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. (2017). Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, Tokio, Japan.

Cettolo, M., Niehues, J., Stüker, S., Bentivoldi, L., Cattoni, R., and Federico, M. (2015). The IWSLT 2015 Evaluation Campaign. In *Proceedings of the Twelfth International Workshop on Spoken Language Translation (IWSLT 2015)*, Da Nang, Vietnam.

Chall, J. and Dale, E. (1995). *Readability revisited: the new Dale-Chall readability formula*. Brookline Books.

Cohn, T. and Lapata, M. (2008). Sentence Compression Beyond Word Deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK. Coling 2008 Organizing Committee.

Dorr, B., Zajic, D., and Schwartz, R. (2003). Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 1–8.

Fevry, T. and Phang, J. (2018). Unsupervised Sentence Compression using Denoising Auto-Encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422, Brussels, Belgium. Association for Computational Linguistics.

Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, 32(3):p221 – 233.

Gal, Y. and Ghahramani, Z. (2016). A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1027–1035, USA. Curran Associates Inc. event-place: Barcelona, Spain.

Ha, T. L., Niehues, J., and Waibel, A. (2016). Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT 2016)*, Seattle, USA.

Ha, T. L., Niehues, J., and Waibel, A. (2017). Effective Strategies in Zero-Shot Neural Machine Translation. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, Tokio, Japan.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M. (2016). Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Kobus, C., Crego, J., and Senellart, J. (2017). Domain Control for Neural Machine Translation. In *Proceddings of Recent Advances in Natural Language Processing (RANLP 2017)*, pages 372–378, Varna, Bulgaria.

Lakew, S. M., Di Gangi, M., and Federico, M. (2019). Controlling the Output Length of Neural Machine Translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, Hong Kong. Zenodo.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*.

Ma, Q., Bojar, O., and Graham, Y. (2018). Results of the WMT18 Metrics Shared Task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Ma, Q., Wei, J., Bojar, O., and Graham, Y. (2019). Results of the WMT19 Metrics Shared Task: Segment-Level and Strong MT Systems Pose Big Challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Marchisio, K., Guo, J., Lai, C.-I., and Koehn, P. (2019). Controlling the Reading Level of Machine Translation Output. In *Proceedings of MT Summit XVII*, volume 1, page 11.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Morristown, NJ, USA. Association for Computational Linguistics.

Pham, N.-Q., Nguyen, T.-S., Niehues, J., Müller, M., Stüker, S., and Waibel, A. (2019). Very Deep Self-Attention Networks for End-to-End Speech Recognition. In *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, Graz, Austria.

Sennrich, R., Birch, A., and Haddow, B. (2016a). Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 35–40, San Diego, California, USA.

Sennrich, R., Haddow, B., and Birch, A. (2016b). 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.

Shimanaka, H., Kajiwara, T., and Komachi, M. (2019). RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 751–758, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112.

Takase, S. and Okazaki, N. (2019). Positional Encoding to Control Output Sequence Length. 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 3999–4004, Minneapolis, Minnesota. Association for Computational Linguistics.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *CoRR*, abs/1706.0.

Wubben, S., van den Bosch, A., and Krahmer, E. (2012). Sentence Simplification by Monolingual Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Zenkel, T., Wuebker, J., and DeNero, J. (2019). Adding Interpretable Attention to Neural Translation Models Improves Word Alignment. *arXiv:1901.11359 [cs]*. arXiv: 1901.11359.