Universität Zürich[UZH]

Bachelorarbeit
zur Erlangung des akademischen Grades
Bachelor of Arts
der Philosophischen Fakultät der Universität Zürich

Sentence Compression in Machine Translation

Verfasser: Lucas Seiler
Matrikel-Nr: 10-755-536

Referent: Prof. Dr. Rico Sennrich

Institut für Computerlinguistik

Abgabedatum: HS22

Abstract

This thesis considers the shortening of a sentence while retaining the essence of the source sentence, so-called sentence compression, not as a monolingual task, but in the context of machine translation. Ideally, a model would be taught the ability to compress sentences using bilingual data with compressed and uncompressed segments aligned. However, monolingual corpora with compression data are already rare, and multilingual compression data is even rarer. We propose three approaches that consist of the specific augmentation and preprocessing of training data, and one approach that introduces a new length penalty within the decoding algorithm. In contrast to the latter approach, which is not very promising in this form, we show that the data-driven approaches can achieve good results even without the availability of multilingual compression data as training data. Moreover, we found that the ability of a translation model to compress sentences does not reduce the translation quality of uncompressed translations, but rather improves them.

Zusammenfassung

Diese Arbeit behandelt die Kürzung eines Satzes unter Beibehaltung von dessen Kerninhalt, die so genannte Satzkompression, nicht als monolinguale Aufgabe, sondern im Kontext der maschinellen Übersetzung. Idealerweise würde einem Modell mittels bilingualer Kompressionsdaten, die sowohl in Originallänge wie auch komprimiert vorliegen, die Fähigkeit beigebracht werden, Sätze zu komprimieren. Monolinguale Korpora, die solche Satzkompressionen beinhalten, sind jedoch rar und mehrsprachige Datensätze dieser Art sind noch seltener. Wir schlagen deshalb drei Ansätze vor, die auf der gezielten Auswahl und Vorverarbeitung von Trainingsdaten basieren, und einen, der eine neue Längenstrafe innerhalb des Decoding einführt. Im Gegensatz zum letztgenannten Ansatz, der in dieser Form nicht überzeugt, zeigen wir, dass die datengetriebenen Ansätze trotz geringer Verfügbarkeit mehrsprachiger Kompressionsdaten gute Ergebnisse erzielen können. Darüber hinaus haben wir festgestellt, dass die Fähigkeit eines Übersetzungsmodells, Sätze zu komprimieren, die Übersetzungsqualität unkomprimierter Übersetzungen nicht senkt, sondern im Gegenteil sogar verbessern kann.

Acknowledgement

I would like to express my gratitude to my supervisor Rico Sennrich for his support and guidance throughout the entire process of completing this thesis. His expertise and encouragement helped me to stay focused and motivated.

I would also like to thank Jonathan Mallinson for his advice regarding the MOSS corpus.

Of course, I would also like to thank my parents for their support throughout my studies.

And finally, I would like to thank my partner for her constant encouragement throughout the whole (and sometimes stressful) journey.

Contents

List of Tables

List of Acronyms

ACL	Association for Computational Linguistics
BPE	Byte Pair Encoding
CR	compression ratio
GPU	Graphics Processing Unit
hyp	Hypothesis
LR	Length ratio
MOSS	Multilingual Compression dataset
MuST-C	Multilingual Speech Translation Corpus
NLP	Natural Language Processing
NMT	Neural Machine Translation
ref	Reference
src	Source
trg	Target
URL	Uniform Resource Locator

1 Introduction

1.1 Motivation

Thanks to deep learning, machine translation is now reaching a level of translation quality that would have been unthinkable just a few years ago. While improving translation quality remains the main topic of machine translation research, the many possibilities offered by neural networks for machine translation are increasingly being explored. These include new features that focus on users influencing the translation output, e.g. by controlling the honorifics of a translation, i.e. setting the desired level of politeness (Sennrich et al., 2016a), by integrating custom terminology into a translation segment at run time (Dinu et al., 2019), or even by simply controlling the preferred language of the translation (Johnson et al., 2017).

In theory, there are countless so-called *constraints* conceivable to control different aspects of machine translation output, whether useful in practice or not. However, one aspect of great practical relevance is controlling the length of machine translation output. Because neural machine translation is prone to generating hypotheses that are too short (Murray and Chiang, 2018), length control has been in the focus of research since the early days of neural machine translation (e.g. Wu et al., 2016)), usually with the goal to generate translations that match the length of a reference translation.

One particular form of length control is sentence compression, which aims at (significantly) shortening or summarising an existing text. While this usually is a monolingual task, when combined with machine translation, sentence compression means that a sentence in one language gets compressed during the translation process. This is called an end-to-end approach, as the translation and compression happen simultaneously. Compared to cascaded process consisting of two or more steps, this approach has the advantage of being faster and less error-prone due to the lack of error multiplication between the indivudal steps.

Two main approaches can be distinguished for this thesis: With the *data-driven approach*, we aim at achieving sentence compression by augmenting the training

data (see Lakew et al., 2019), and with the *decoding approach*, the goal is to generate compressed sentences by modifying the decoding algorithm (see Niehues, 2020).

Assessing sentence compression across language boundaries poses some difficulties: Firstly, multilingual corpora containing sentence compression are very rare. Secondly, while the subword is generally the smallest unit in neural machine translation (NMT), the number of letters is much more appropriate as a metric to measure the length of a sentence.

1.2 Research Questions

The objective of this thesis is to answer the following research questions:

1. Can a successful approach be found to develop a data-driven model for sentence compression in spite of the limited availability of multilingual sentence compressions?
2. How promising is a decoding approach that focuses solely on modifying the decoding algorithm during inference?

1.3 Thesis Structure

This first chapter served as an introduction to the topic and the objectives of this thesis. Chapter 2 contains a review of previous work and highlights the challenges of multilingual sentence compression. Chapter 3 provides an overview over the different corpora that are used as training and evaluation data for the experiments described in Chapter 4. The results of these experiments are presented in Chapter 5, followed by a discussion of the results in Chapter 6. The final chapter, Chapter 7, brings the thesis to a close.

2 Background

2.1 Sentence Length in Machine Translation

Generating a translation with the correct length has been a challenge since the early beginnings of NMT. This phenomenon is called *length bias* and stems from the fact that the hypotheses generated by NMT systems are, on average, shorter than their references (Müller and Sennrich, 2021). The preliminary reason for this is that during beam search, the probability of each (additional) token in a hypothesis is multiplied with the probability of the hypothesis. This leads to a lower probability of longer hypotheses compared to shorter ones, hence shorter hypotheses are, on average, favoured (Wu et al., 2016). Therefore, a technique called *length normalisation* was introduced by Wu et al. (2016) to mitigate the length bias, and nowadays, NMT toolkits such as Sockeye (Hieber et al., 2022) contain a variety of mechanisms and hyperparameters to ensure the correct translation length.

Within the context of the length bias, the length of a translation is compared to that of its reference. However, research has also been conducted to match the length of the source text. This comparison is usually not of interest, but under certain circumstances it is desirable for the translated segment to be of the same length (or shorter) as the source segment. If, for example, a certain layout has to be adhered to or in the case of subtitles, only a limited number of characters are available. Under these circumstances, we would like to obtain translations that match length of the source text, even though this may mean that we generate a translation that is shorter than its reference. Since in these use cases the complete information of the original sentence should be represented in the translation, it is noteworthy that this is not a form of sentence compression.

Lakew et al. (2019) and Niehues (2020) both proposed an approach to maintain the length of a source segment in the translation process: While Lakew et al. (2019) combines a data-driven approach with the integration of length information into the positional embedding of the transformer, Niehues (2020) is integrating various additional constraints into the model to make it length aware.

One key difference between the two approaches is that Lakew et al. (2019) evaluates the sentence length on character level, whereas Niehues (2020) measures the segment length in number of subword tokens. Furthermore, while Lakew et al. (2019) only uses soft constraints, Niehues (2020) also restricts the search space to only include hypotheses that exactly match the desired length. The difference between these approaches can be seen in the results: While the hard constraint by Niehues (2020) seems to substantially lower the translation quality, the results by Lakew et al. (2019) look promising.

2.2 Constraints in Machine Translation

In general , the term *constraint* refers to a certain requirement that should be fulfilled during the translation process. Constraints can take various forms and enable the user, for example, to integrate or avoid certain words in a generated target sentence using *lexical constraints* (Jon et al., 2021). Other types of constraints allow the user to select the politeness level of a translation (Sennrich et al., 2016a) or to choose either the active or passive voice for a target sentence (Yamagishi et al., 2016).

Two main types of constraints can be distinguished: *hard constraints* and *soft constraints*. Hard constraints are guaranteed to be achieved, while soft constraints try to nudge a model into the fulfilment of the constraint, but can also be ignored. In short, hard constraints enforce a certain constraint, while soft constraints merely try to provoke the model to be fulfilled.

Matching a certain requirement can often be achieved by both hard and soft constraints, both of which usually have their advantages and disadvantages. For example, both soft and hard constraints have been implemented to allow lexical constraints to be integrated into machine translation. While *Grid Beam Search* by (Hokamp and Liu, 2017) guarantees that a certain term is inserted into a translation, this approach results in a slow decoding speed. The approach by (Dinu et al., 2019) on the other hand does not guarantee that a certain term is inserted into a translation, but does not impact decoding speed. This shows that choosing the right type of constraint is usually a trade-off between several factors.

2.3 Sentence Compression as a Task

Sentence compression is the task of shortening a sentence while preserving the essence of the source sentence. Two main approaches can be found: The deletion-based approach is based on omitting unncessary words, whereas the abstractive approach involves rewriting the compressed sentence from scratch, therefore substituting words and rephrasing the entire sentence (Yu et al., 2018). As such, sentence compression has been a standard NLP task for a long time (Filippova et al., 2015).

Following Pitler (2010), a compressed sentence should fulfil the following three requirements[1]:

It should

- be shorter than the source sentence
- preserve the most important information of the source sentence
- itself be grammatical.

Sentence compression has traditionally been a monolingual task as part of different fields in NLP, such as text summarisation and text simplification. While in text summarisation, the goal is to summarise a certain sentence or document, thereby reducing its length while maintaining key information (Filippova, 2010; Roy, 2020), in text simplification, sentence compression is just a means to an end: By reducing the length of the text through the removal of non-essential information, the text becomes more accessible to people with reduced literacy (Angrosh et al., 2014).

The potential of encoder-decoder models (Sutskever et al., 2014) was also applied on monolingual sentence compression: Kikuchi et al. (2016) evaluated methods to control the length of encoder-decoder output in a text summarisation task, using two learning-based and two decoding-based approaches. The key difference between these two methods is that the learning-based method receives a requested length as input, while the decoding-based method only receives this crucial information during the decoding process. The results showed that the learning-based methods were able to control length without affecting the summary quality in a text summarisation task.

[1] See also the following instructions for human annotators to create valid sentence compressions: `https://homepages.inf.ed.ac.uk/mlap/cgi-exp/annotators1.html`

2.4 Sentence Compression in Machine Translation

In contrast to sentence compression discussed so far, sentence compression in machine translation is not a monolingual task, but a multilingual one. According to our definition of sentence compression in Section 2.3, sentence compression occurs only when a translated sentence is shorter than the source sentence. Thus, methods for length control, such as the approach by Lakew et al. (2019), which aim at matching the length of the source sentence, do not count as sentence compression.

Approaches to multilingual sentence compression are difficult to find in research: Mallinson et al. (2018) applied a combination of machine translation and a length control approach similar to Kikuchi et al. (2016) to achieve monolingual sentence compression. By controlling the sentence length during the translation process to and from a second language (bilingual pivoting), sentence compression was achieved in English, French, and German and for different compression ratios. While the approach proved successful in general, it only performed well up to a certain degree of compression. Li et al. (2019) proposed to enhance the source sentence representation within a translation model using a compressed version of said sentence, thereby providing the model with the gist of the source segment. The objective of this approach was not to generate compressed translations, but to improve the overall translation quality. Niehues (2020) successfully applied his approach for length control described in Section 2.1 also on sentence compression.

3 Data

3.1 Parallel Data

Following Lakew et al. (2019), we chose to to use MuST-C[1] (Di Gangi et al., 2019), a corpus containing multilingual speech translations based on English TED talks. We utilised the data available for the language pair English–German for the language direction German–English. The data in the corpus came divided into a training, validation, and test set as shown in table 1. We left this split of the data set unchanged. In line with the preprocessing steps described in Lakew et al. (2019), we tokenised the data with the Moses tokeniser (Koehn, 2005) and learnt byte-pair encoding (BPE; Sennrich et al., 2016c) on the training data on a joint vocabulary with 32,000 merge operations. This joint vocabulary was then used to split the tokenised data into subwords.

	Train	Val	Test	Total
MuST-C De-En	229,703	1,423	2,641	233,767

Table 1: Number of training, validation, and test segments in the MuST-C corpus.

To create our back-translations (Sennrich et al., 2016b), we trained an auxiliary En-De model with significantly more data to increase the translation quality of our back-translations. For this, we used the English-German data of the following three corpora: Common Crawl,[2] Europarl v7,[3] and News Commentary v9.[4] A strict ratio filtering with a maximum length ratio of 1.5 was applied to remove potentially low-quality segments, which left us with a total of 3,898,156 segments,

[1]Publicly available for download at `https://ict.fbk.eu/must-c`

[2]Publicly available for download at `http://www.statmt.org/wmt13/training-parallel-commoncrawl.tgz`

[3]Publicly available for download at `http://www.statmt.org/wmt13/training-parallel-europarl-v7.tgz`

[4]Publicly available for download at `http://www.statmt.org/wmt14/training-parallel-nc-v9.tgz`

divided into 3,895,656 training, 1,000 validation, and 1,500 test segments. Besides the additional filtering step, we applied the same preprocessing steps (tokenisation as well as learning and applying BPE) as described for the MuST-C corpus.

	Unfiltered	Filtered
Common Crawl De-En	2,399,123	1,895,975
Europarl v7 De-En	1,920,209	1,814,427
News Commentary v9 De-En	201,288	88,347
Total	4,520,620	3,798,749

Table 2: Number of sentence pairs to train the auxiliary model.

3.2 Monolingual Data

Corpora containing parallel data with uncompressed and compressed sentences are very scarce. For our experiment with compressed data, we used the monolingual corpus built by Filippova and Altun (2013).[5] This corpus is by far the largest one in the field of sentence compression. The approach behind this automatically created corpus is to leverage the relationship between headlines of news articles and their first sentences. Given a sufficient similarity, this can be used to extract a compression of the sentence. We extracted both the uncompressed and compressed parallel sentences. While Filippova and Altun (2013) note that the corpus contains 250,000 parallel segments, we were only able to extract a total of 210,000 segments.

	Train	Val	Total
Sentence Compression corpus	200,000	10,000	210,000

Table 3: Number of parallel sentences in the Sentence Compression corpus.

3.3 Test Sets

Compared to corpora containing monolingual compression data, bilingual data where the uncompressed and compressed data are held in a different language are even more

[5]Publicly available for download at
`https://github.com/google-research-datasets/sentence-compression`

difficult to find. However, this kind of data is essential to properly evaluate sentence compression in the context of automatically translated text.

One such corpus is MOSS (Mallinson et al., 2018), a multilingual parallel corpus containing documents in English, French, and German. The documents stem from the European parliament proceedings, TED talks, news commentaries, and the EU bookshop. Each document comprises 100 segments. For each language, five native speakers created one compression for each of the four documents, leading to 2,000 compressed sentences per language aligned with their source sentences. It should be noted that the documents in the three languages are not aligned at the sentence level, as one German sentence could be aligned to multiple English sentences (or vice versa) (J. Mallinson, personal communication, 7 September 2022).

We downloaded the corpus[6] and reviewed both the English and German compressions:

- *Eubook*: The 5 compressions of the German version all showed repeated errors: omission of necessary nouns, orthographic errors (including lack of upper and lower case), and misalignments between the original document and the compression. The English version also shows some shortcomings, especially in the compressed reference labeled *task1_ref1*, which not only contains linguistic errors, but also partly misrepresents the meaning of the source segment. Compared to the German version, however, the shortcomings overall are more minor. As the documents are not aligned on the sentence level, we translated the original (uncompressed) English document from English into German. As there were only a total of 100 segments to translate from English into German and translation quality is of the upmost essence in this case, we used a commercially available translation model[7] to create a back-translated version of the document.

- *Europarl*: In the German versions, there were minor punctuation errors, as well as other minor punctuation errors in every text. However, *task1_ref4* was bursting with spelling errors, missing capitalisation and content errors, so we decided to use only the other documents. Interestingly, *task1_ref4* for English was significantly shorter than the other compressions and had repeated missing punctuation. However, because the compressions seemed accurate, this reference document was not removed. As for Eubooks, we again created back-translated version of the original English document.

[6]Publicly available for download at `https://github.com/Jmallins/MOSS`

[7]Model developed by the Swiss Machine Translation company TextShuttle AG (`https://textshuttle.ai`.

- *News*: By and large, both the English and the German versions were acceptable, except for the German *task1_ref4*, which again showed the problems already known from the Europarl document. Although the English and German versions are not aligned, the same sentences can be found in both versions, only in different order. Therefore, we managed to create a sentence-based alignment by manually rearranging the German version.
- *Ted*: Interestingly, in both the English and German versions, except for very short sentences, each sentence was spread over several lines. A document with such segmentation is not suitable as input for a machine translation model, which is why we omitted this document from the test set.

	Eubook	Europarl	News
Reference 0	0.79	0.7	0.74
Reference 1	0.77	0.63	0.77
Reference 2	0.78	0.74	0.74
Reference 3	0.79	0.7	0.74
Reference 4	0.78	0.41	0.7
Average	0.78	0.64	0.79

Table 4: Compression ratios for the MOSS corpus in English.

	Eubook	Europarl	News
Reference 0	0.69	0.62	0.6
Reference 1	0.68	0.57	0.63
Reference 2	0.69	0.66	0.6
Reference 3	0.69	0.62	0.6
Reference 4	0.68	0.37	0.57
Average			

Table 5: Compression ratios for the compressions in the MOSS corpus in English, compared to the uncompressed German versions (Europarl and Eubook are back-translations).

As we can see from Tables 4 and 5, the compression ratio varies between each domain and annotator, and also substantially between monolingual (English–English, Table 4) and multilingual (German–English, Table 5) sentence compression. For the Eubook document, however, the compression ratios between the individual annotators are surprisingly similar. We verified this and could not detect any irregularities.

4 Methods

4.1 General Model Architecture and Hyperparameters

All models trained for this thesis were based on the Transformer architecture (Vaswani et al., 2017). We followed Lakew et al. (2019) for the hyperparameter settings, but made certain adjustments to the configuration to increase training speed. All of our models consisted of 6 encoding and decoding layers, 8 attention heads, a hidden/embedding size of 512, and a feed forward layers size of 2048. This corresponds to the Transformer Base Model configuration in Vaswani et al. (2017). The batch size was set to 4500 tokens. All models are trained on one NVIDIA GeForce GTX TITAN X GPU.

We used two toolkits: Sockeye 3 (Hieber et al., 2022) and Joey NMT (Kreutzer et al., 2019). A key difference between the two toolkits is that Sockye supports factored NMT, whereas Joey NMT does not. For the Sockeye models, we used early stopping with validation perplexity as metric to end the training after no improvement for 8 checkpoints. Our Joey NMT models train for a fixed duration of 50 epochs. With both toolkits, we used the default settings for all hyperparameters not specified above, including learning rate.

We trained a total of seven models:

- A German–English Sockeye baseline model, referred to as *Sockeye Baseline*
- A German–English Joey NMT baseline model, referred to as *Joey NMT Baseline*
- A German–English Sockeye model with length tags, referred to as *LengthTag*
- A German–English Sockeye model with length factors, referred to as *LengthFactors*
- A German–English Sockeye model with compression tags, referred to as *CompTag*
- A German–English Sockeye model with compression tags and length factors, referred to as *CompFactors*
- A English–German Sockeye model for back-translations

4.2 Data-driven Approaches

The idea behind our data-driven approach was to leverage data to teach a model to shorten a source segment while translating it. All of the following approaches were based on the model learning from existing (un)compressed data. The focus lied therefore on the selection of training data and the necessary preprocessing steps; changes to the decoding algorithm were not necessary.

4.2.1 Annotation of Training Data with a Length Tag

As we have seen, corpora with long-short sentence pairs are very rare. However, we have a large amount of bilingual data. It is the intrinsic nature of translation that the length ratio between a source sentence and its translation varies. We can therefore try to use this variation in length ratio to shorten sentences.

As our first data-driven approach, we implemented a variaton of the *length token method* described in Lakew et al. (2019).

We followed Lakew et al. (2019) and divided our training data into three distinct length categories, each with a specific length ratio. Lakew et al. (2019) set the length ratios for the categories to [0,1], [1,1.2], and (1.2,∞) for the language directions English–Italian and English–German. In order to encourage shorter translations, we defined lower ratio bounds. Of course, the more the length ratio deviates from the natural length ratio of a language direction, the more difficult it becomes to find matching segments in the training data. However, in contrast to the two language directions chosen by Lakew et al. (2019), English translations tend to be shorter than their German source sentences. This allowed us to choose lower length ratios overall.

The first category *short* consisted of source-target pairs where the target segment was significantly shorter than the source segment. As we were just exploiting the natural variance in length ratios, we hypothesized that the target segments in this category would still fully cover the content of the source segment and would therefore not be a compressed version of the source segment, but just short translations. The second category *normal* contained parallel segments where the target side was only slightly shorter than the source side, and the third category *long* contained the segments where the target segment was longer than the source segment.

As training data, we used the MuST-C corpus described in section 3.1. We annotated the training data by prepending the appropriate length tag (depending on

	Set	Short	Normal	Long
	train	41,164	123,098	65,441
	val	295	826	302
	test	516	1,542	583
Length ratio		[0,0.8]	(0.8,1]	(1,∞)

Table 6: Training, validation, and test set divided into each length ratio category (measured by number of characters).

the length ratio between each target and source) to each source segment. Once the training data was tagged, the training was carried out normally. During inference, we were then able to set the approximate length of the text translation by adding the corresponding tag to the segment to give the model an indication of the desired length ratio.

				Length ratio
Source	`Go away !`	Target	`Hau ab !`	0.88
Source with tag	`<short> Go away !`			
Source	`Go away !`	Target	`Verschwinde !`	1.5
Source with tag	`<long > Go away !`			

Table 7: Example of length tag prepended to source segment for training.

4.2.2 Back-Translations with Compressed Segments

We have described a large monolingual corpus with uncompressed and compressed sentence pairs in section 3.2. In order to train a machine translation model, though, bilingual data is required. Thankfully, it is possible to leverage monolingual data for machine translation by automatically translating the future target side of the training data into the source language in a process called *back-translation* (Sennrich et al., 2016b; Edunov et al., 2018; Caswell et al., 2019). This way, we can generate synthetic training data that we can use alongside the original training data. Of course, an existing translation model (in this case for English–German) is a prerequisite for this.

It has been shown by Edunov et al. (2018) that adding noise to the back-translations

by sampling from the model distribution during back-translation outperforms regular beam search, because this way, a stronger training signal is provided to the model. Caswell et al. (2019) argued, though, that this added noise in the synthetic data simply serves as an indicator to a model trained with back-translations that a certain source segment stems from synthetic data. For that reason, they proposed adding a `<BT>` tag to the back-translations to signal to the model that a segment was back-translated. As we show below, we labeled the segments generated via back-translations with a compression tag, which implicitly also served as a label indicating back-translation. Hence, we refrained from sampling and used regular beam search (with beam size 5) to create our back-translations from the Sentence Compression corpus. The model used for this was an auxiliary model for English–German trained on significantly more data described in Table 2 than our other models to ensure better translation quality.

	Train	Val	Test
Sentence Compression corpus	200,000	750	1,500
MuST-C corpus	200,000	750	1,500
Total	400,000	1,500	3,000

Table 8: Composition of training data for *CompTag*.

Instead of using only the back-translations to train our compression model, we also added the MuST-C corpus to the training data. This served two purposes: Firstly, we wanted our model to not only create compressions while translating, but also to be able to translate sentences without shortening them. Secondly, Caswell et al. (2019) brought forward that back-translations amplify the already existing underlying biases in machine translations. These biases could be detrimental to the overall translation quality, which is another reason for including non-synthetic data. We used a 1:1 ratio of back-translated compression data to segments from the MuST-C corpus.

Source	`<U> Der Hund, der den Nachbarn gehört, hat mich gebissen .`
Uncompressed target	`The dog that belongs to the neighbors barked at me .`
Source	`<C> Der Hund, der den Nachbarn gehört, hat mich gebissen .`
Compressed target	`The dog barked at me .`

Table 9: Example of expected behaviour of the *compress* and *uncompress* tag.

We then annotated the data with a tag labeling if the target side of a parallel

segment was a back-translation segment and compressed (<C>) or came from the MuST-C corpus and was therefore uncompressed (<U>). Similar to the approach using length tags in subsection 4.2.1, we tried to train the model to compress a sentence by prepending the <C> tag to a source segment. Choosing the appropriate compression ratio for each segment was left to the model.

4.2.3 Target Factors with Token Length

Factored machine translation is a method that allows us to incorporate additional input and output streams into a model. This way, the model receives additional information for each token. Depending on whether these streams are provided on the source or target side, they are either called *source factors* or *target factors*. In NMT, factors can be used to integrate various different kinds of information, such as linguistic features (Sennrich and Haddow, 2016), terminology integration (Dinu et al., 2019), and casing information (Etchegoyhen and Gete, 2020).

Our approach was to make a model aware of the length of each subword token in the target segment by providing this information during training as *length factors* on the target side for each target segment. During training, the model should learn the length of each subword. We then added a tag on the source side that contained the sum of all length factors, i.e. the length of the entire target segment (measured in characters). This way, the model should learn to make the link between the length token on the source side and the sum of the length factors on the target side. We hypothesised that during inference, the model would know from the tag how long the target segment should be.

When calculating the length of our subword tokens, we cannot just count the subwords themselves, because then the spaces would not be considered. We also need to be mindful of the BPE word divider symbol @@ that indicates that the subword is at the beginning or in the middle of a word and is therefore not followed by a space. We follow these steps to determine the length of a subword:

- We calculated +1 to the length of each subword to account for the subsequent space, except in the following two cases:
  - The symbol @@ was not counted, and no space was added to the calculation of the subword length.
  - We also did not add the space to the length of the last subword of a segment (usually a period).

For punctuation marks such as commas and semicolons in the sentence, the segment length contained spaces that did not actually exist in the source segment. As we tried to achieve (significant) compression, this was not a problem, however, because it would only result in the generated translation being at most a few characters shorter.

Source	<31> The price of natural gas has increased .						
Target	Der	Erd@@	gas@@	preis	ist	gestiegen	.
Length factors	4	3	3	6	4	10	1

Table 10: Example of length factors for each subword token in the target segment.

We used the MuST-C corpus, created length factors for the target side and annotated the source side with a tag containing the target length. We also trained a German–English model that combined this approach with the back-translations approach described in subsection 4.2.2. We were able to use the data that had been annotated with the compression tag to create the corresponding length factors. On the source side, this meant that there were two tags at the beginning of each source segment: one tag that entailed the target segment length and another other tag that represented the compression tag (e.g. `<115> <U>`).

4.3 Constrained Decoding Approach

In contrast to the data-driven approaches, our constrained decoding approach tries to modify the decoding algorithm to generate shorter hypotheses. Thus, the compression is not learned from training data. In the present case, we only tried to make changes to the beam search so that a model training or finetuning would not be required. Hence, even existing models with the proposed modifications would be able to generate sentence compressions without having to be retrained or fine-tuned.

A model based on the transformer architecture is not aware of translation length. As we have shown in 2.1, this is why achieving the right target length has been a research topic since the early beginnings of NMT. Lakew et al. (2019) and Niehues (2020) have shown that one way to achieve length-awareness in a transformer architecture is to modify the positional encoding, with some kind of training involved in both methods. Therefore, mere modifications to beam search to create length awareness within the model are not technically feasible, as length awareness of a model would require new training or at least fine-tuning.

Computing the all possible hypotheses $|V|^L$ during the generation of a translation, with $|V|$ being the target vocabulary and L being the maximal sequence length, is intractable (Hokamp and Liu, 2017). Greedy decoding as an alternative decoding strategy selects the subword with the highest probability at each time step, but does not lead to a globally optimal solution. As a compromise between these two strategies, beam search only keeps k hypotheses at each time step, k being the beam size. At each time step, the k partial hypotheses are extended with every possible element of the vocabulary, and only the best k hypotheses are kept. This continues for each of the hypotheses until the end-of-sequence symbol is generated (Sutskever et al., 2014). At each timestep, the activation function *Softmax* in the last layer of the neural network outputs a probability distribution over the target vocabulary at each time step. This distribution serves as the basis for adding up the probability of each newly added token to the probability of the beam.

We hypothesized that modifying this probability distribution to favour shorter subwords enables a model to generate shorter hypotheses. To achieve this, we introduced a *length penalty* for each subword in a vocabulary V depending on its length: By dividing the probability $p(y)$ of each subword y where $y \in V$ with its length l so that with each character in a subword, we lowered the probability of generating it. This results in the modified probability $p_{pen}(y)$ for a subword with each additional character that it contains. When calculating the length of each subword, the word divison symbol `@@` was disregarded. The probability of special symbols such as the end-of-sequence symbol was left unmodified.

Furthermore, we the probabilities of subwords that contained the BPE tag `@@` and therefore are at the beginning or in the middle of a word to $-\infty$, thereby avoiding complex words consisting of several subwords.

The following experiments were conducted with the Joey NMT toolkit Kreutzer et al. (2019):

1. *No length normalisation*: In this experiment, we set the hyperparameter for length normalisation α to 0, thus deactivated it. We named it in the evaluation *No length norm.*

2. *Linear length penalty*: We introduced a linear length penalty $p_{pen}(y) = p(y)/l$ s.t. $l \geq 1$ and referred to this experiment as *LP linear*.

3. *Linear length penalty & no BPE tags*: Besides applying a linear length penalty, we also avoid the generating of subwords containing BPE tags. This model named *LP linear & no BPE tags*.

4. *Exponential length penalty*: We modified our linear length penalty by making it exponential with l being the exponent of e: $p_{pen}(y) = p(y)/\exp(l)$ s.t. $l \geq 1$ and refer to this model as *LP linear*. We referred to this experiment as *LP exp.*

5. *Exponential length penalty no BPE tags*: This experiment was a combintion of *LP exp.* and the suppression of BPE tags. This experiment was called *LP exp. & no BPE tags*.

4.4 Evaluation Setup

In line with our research questions, we evaluated each experiment according to two main points:

1. What compression ratio is the model able to achieve?
2. What is the translation quality for compressed and uncompressed translations?

We evaluated on the following metrics:

- BLEU: BLEU (Papineni et al., 2002) is still one of the most widely used metrics to automatically evaluate translation quality. BLEU measures the similarity between a hypothesis and one or multiple references based on n-gram matches. We measured BLEU with the sacreBLEU implementation[1] by Post (2018).
- BLEU*: Because BLEU is precision-based, hypotheses that are shorter than the reference would have an advantage. To offset this, BLEU includes a *brevity penalty*, which penalizes hypotheses that are shorter than their references. In the case of compressed sentences, however, it is desirable for the hypotheses to be shorter than their references. Inspired by Lakew et al. (2019), we therefore deactivated the brevity penalty and called this modified BLEU version BLEU*.
- Length ratio: We measured the average length ratio both between hypothesis and source (LR^{src}) and between hypothesis and reference (LR^{ref}).
- Avrg. chars/word: To evaluate our decoding approach, we reported the average number of characters per word.

We used the test sets described in Section 3.3 and, when evaluating on the MOSS test set, compared each hypothesis to all available references of a document. The results for the data-driven and the decoding based approaches were kept separate.

[1]Publicly available for download at `https://github.com/mjpost/sacrebleu`

We also trained a baseline with each toolkit with the hyperparameters set out in Section 4.1. For all tranlations, beam size was set to 5.

5 Results

5.1 MOSS Test Set

We evaluated our models according to the evaluation setup proposed in Section 4.4 and discuss the model performances on three German–English documents from the MOSS corpus. It is worth to note that Europarl and Eubook are back-translations. We report for each model the constraint that has been given to it as input. Because we use multiple references, LR^{ref} is an average of the length ratio between the hypothesis and each reference. Input for sacreBLEU is provided as detokenised text.[1].

MOSS News De-En					
Model	Constraint	BLEU	BLEU*	LR^{src}	LR^{ref}
LengthTag	short	21.2	21.2	0.77	1.40
Baseline	truncated	80.5			
CompTag	compression	18.3	**24.4**	**0.46**	**0.71**
Baseline	truncated	50.2			
LengthFactors	CR = 0.75	19.9	19.9	0.85	1.54
Baseline	truncated	81.8			
LengthFactors	CR = 0.5	20.5	20.5	0.61	1.11
Baseline	truncated	78.8			
CompFactors	comp. & CR = 0.75	**23.2**	23.2	0.82	1.49
Baseline	truncated	81.2			
CompFactors	comp. & CR = 0.5	22.7	22.7	0.57	1.03
Baseline	truncated	75.9			

Table 11: Results for the evaluation of the data-driven approaches on the MOSS News document.

For the evaluation on the News document, the results in Table 11 show that all models have achieved some level of sentence compression, varying from a very low

[1] nrefs:5|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1

compression ratio of 0.85 by LengthFactors compared to 0.46 by CompTag. The latter model is also the only one that has succeeded in outperforming the average compression ratio of the reference documents. Looking at the BLEU scores, it is immediately noticeable that the baseline, in which each segment is shortened to the length of the hypothesis of each model, was significantly better than the hypotheses. This will be discussed in Chapter 6. With regards to the translation quality of the models, CompTag is back to the top: although its BLEU score is the lowest, this is due to the strong compression ratio and the brevity penalty associated with it. We also see that with BLEU*, CompTag scored the highest.

The LengthTag model showed a solid result with a compression ratio of less than 0.8. This is the threshold below which the training data for the short category had to be. Interesting are the values for the Length Factor model, which received two different compression ratios as input: Although the use of target factors seems to influence length, the compression is below the desired compression. Finally, the results show that CompFactors, a combination of the approaches for LengthFactors and CompTag, generated a compression ratio that lied between the results of the two approaches, but achieved the best result on BLEU.

MOSS Europarl De-En					
Model	Constraint	BLEU	BLEU*	LR^{src}	LR^{ref}
LengthTag	short	24.1	24.1	0.79	1.87
CompTag	compression	**35.0**	**36.8**	**0.44**	**0.93**
LengthFactors	CR = 0.75	26.0	26.0	0.86	2.04
LengthFactors	CR = 0.5	24.1	24.1	0.61	1.45
CompFactors	comp. & CR = 0.75	30.8	30.8	0.8	1.91
CompFactors	comp. & CR = 0.5	31.2	31.6	0.54	1.28

Table 12: Results for the evaluation of the data-driven approaches on the MOSS Europarl document.

The results of the Europarl evaluation are very similar to those of Table 11. CompTag was able to achieve the highest BLEU score while simultaneously having generated the translation with the highest compression rate. The results also shows again that LengthFactors failed to match the expected the compression ratio and that the compression achieved is at the expense of translation quality. LengthTag scored the

worst on the Europarl document.

MOSS Eubook De-En					
Model	Constraint	BLEU	BLEU*	LR^{src}	LR^{ref}
LengthTag	short	22.5	22.5	0.78	1.27
CompTag	compression	24.9	**29.3**	**0.55**	**0.85**
LengthFactors	CR = 0.75	22.1	22.1	0.89	1.44
LengthFactors	CR = 0.5	22.5	22.5	0.66	1.07
CompFactors	comp. & CR = 0.75	**27.9**	27.9	0.82	1.33
CompFactors	comp. & CR = 0.5	26.1	27.9	0.56	0.91

Table 13: Results for the evaluation of the data-driven approaches on the MOSS Eubook document.

The evaluation on the Eubook document resembles that of the other two documents. To sum up the evaluation on the MOSS test set, it can be said that with CompTag, only the approach that was only trained on compressed sentence without other constraints, consistently undercut the length of the reference compressions. Although CompFactors also was trained on compressed data and surpasses CompTag on BLEU in two occasions, it falls short regarding the compression ratio.

5.2 MuST-C Test Set

The evaluation on the MuST-C test set serves as an assessment of the general (uncompressed) translation quality. Therefore, the models only receive input constraints that should not result in compressed sentences. Furthermore, because we want full coverage of the content source segment, we refrain from evaluating on BLEU*.

For LengthFactors and CompFactors, the constraint was to generate translations of the same length as the source segment. However, in an additional oracle experiment, each constraint was chosen according to the length of each target segment; hence, the models were aware of the reference length. Unsurprisingly, this information helped to ensure that the models not only matched the reference length better than the other models, but also led to them surpassing the baseline in terms of BLEU.

Five out of eight models managed to exceed the baseline. In the case of the other three (LengthTag, LengthFactors and CompFactors), the constraint caused the generation of too long translations, which is a disadvantage when calculating BLEU. Somewhat surprisingly, CompTag also scored best on the uncompressed test set.

MuST-C Test Set De-En				
Model	Constraint	BLEU	LR^{src}	LR^{ref}
Sockeye baseline		33.0	0.87	0.96
LengthTag	oracle	34.0	0.87	0.96
LengthTag	normal	33.8	0.87	0.96
LengthTag	long	31.9	0.95	1.06
CompTag	No compression	**36.2**	0.88	0.97
LengthFactors	oracle	35.8	0.89	**0.98**
LengthFactors	CR = 1	30.2	1.02	1.14
CompFactors	No comp. & oracle	35.7	0.90	**0.98**
CompFactors	No comp. & CR = l	30.2	**1.01**	1.13

Table 14: Results for the evaluation of the data-driven approaches on the MuST-C test set.

The results for our constrained decoding approaches on the identical test set show that the Joey NMT baseline model was able to achieve a rather unexpected strong length ratio compared to the source text of nearly 1, thereby generating longer translations than the reference. We can also see that deactivating length normalisation by setting α to 0 helped us to generate shorter translations, while simultaneously lowering the BLEU score by -0.4.

The introduction of a length penalty resulted in a decrease of translation quality, while the sentences were not significantly shorter compared to the result with deactivated length normalisation. we also report that the exponential calculation of the length penalty had a greater effect, which, in return, significantly reduced the BLEU score compared to the linear penalty and, by avoiding the use of several subwords, more than halved the BLEU score (30.1 vs. 14.6). The "Average character per word" metric shows that avoiding BPE tags resulted in generating shorter words, with the average word length being reduced from 4.8 characters (Joey NMT Baseline) to 3.44 words (LP exp. & no BPE tags).

MuST-C Test Set De-En				
	BLEU	lrsrc	lrref	Avrg. chars/word
Joey NMT Baseline	**35.8**	**0.99**	1.08	4.8
No length norm. ($\alpha = 0$)	35.4	0.85	**0.95**	**4.81**
LP linear	34.1	0.83	0.92	4.66
LP linear & no BPE tags	30.1	0.84	0.93	4.4
LP exp &	19.5	0.75	0.83	4.16
LP exp. & no BPE tags	14.6	0.79	0.87	3.44

Table 15: Results for the evaluation of the constrained decoding approach on the MuST-C test set.

6 Discussion

6.1 Data-driven Approaches

The reported length ratios LR^{src} as part of the evaluation of the MOSS corpus show that each model that was trained with modified training data was able to reduce the length of the source sentences – albeit with different compression ratios.

Our results for LengthTag are in line with the findings by Lakew et al. (2019) on adding length tags. The set length ratio ≤ 0.8 for the *short* token was reproduced during inference on all three documents of the MOSS corpus (0.77, 0.79, and 0.78). But because we had annotated a normal corpus where the source and the target page were supposed to contain the same content, it is unclear whether the target segments are actually compressed, or just short translations that still fully cover the content of the source sentence. Because of the nature of the training data, we suspect that the translations are short but complete. An analysis of translation samples supports the hypothesis that the translation produced with the *short* constraint still fully represent the meaning of the source segment.

Regarding the evaluation of our approach using length tags, the results are mixed: Although the two constraints to generate translations with a compression ratio of 0.75 and 0.5 each had an impact on the length of the generated translation, they miss the desired length ratio by at least 0.1 (e.g. 0.85 instead of 0.75 and 0.61 instead of 0.5 on the News document). Although one could argue that this is because the LengthFactors did not see any evidence of compressions in training, this argument falls short: CompFactors, whose length factors were trained with back-translated compressions, also failed to meet the given compression ratios. This suggests that this is not due to a data problem. However, LengthFactors and CompFactors performed well on the uncompressed MuST-C test set: Here, they almost reached the given reference in the oracle experiment, and the constraint to generate sets of the same length as the source was best met by both models with LR^{src}=1.02 (LengthFactors) and LR^{src}=10.2 (CompFactors), respectively.

The approach of using existing compressions as back-translations and tagging them

was the most successful approach in terms of compression ratio. This is not surprising, since this model had seen examples of compressed sentences during training. However, the compression ratio between the back-translated uncompressed side of the Sentence Compression corpus and the compressed original side shows a compression rate of 0.36. The compression ratio of CompTag on the MOSS corpus are slightly higher and vary (0.46, 0.44, and 0.55). This raises the question of whether the model had learned which compression level to apply. We can see from the LR^{ref} of 0.71 for the News document, that this not the case, as the generated translation is significantly shorter than the average of the five references. With regards to the good performance of this model on BLEU, one simple reason could be that CompTag saw almost twice as much training data as the other models, resulting in these above-average BLEU scores.

We hypothesize that further research on the use of regular bilingual data may lead to improvements on multilingual sentence compression without integrating actual compressed data into the training data with an approach that Niehues (2020) calls "pseudo-supervised training".

6.2 Constrained Decoding Approach

The results for the experiment with deactivated length normalisation are in line with expectations and show a length ratio 0.15 lower than the reference. In contrast, the results for the influence of the implemented length penalty are surprising, as the influence on the length ratio is less than expected. This suggests that the probability distribution after the Softmax activation function was very certain about the probabilities of the appropriate next tokens in the vocabulary. This is also evident in the different results for the linear and exponential length penalty (without avoiding BPE tags) (0.83 vs. 0.75): The exponential length penalty had a much higher impact on avoiding the generation of longer words.

The findings also show that avoiding the generation of words consisting of several subwords, i.e. words with BPE tags, did not result in shorter translations, but only in the generation of shorter words, as shown by the exponential length penalty with and without the avoidance of BPE tags (avrg. chars/word 4.16 vs 3.44). In fact, with regard to translation length, this seems to have the opposite effect; translations became slightly longer when avoiding BPE tags. This suggests that a model does not generate a shorter term as an alternative fora word composed of several subwords, but rather paraphrases an alternative that consists of more characters, leading to a longer (and in terms of quality much worse) translation.

Creating length-awareness within a model does not seem possible at present without at least some form of training that allows the model to learn the relationship between a source and target segment and their lengths. A length penalty purely based on subword length only seems to compromise translation quality without achieving the desired compression effect and, depending on the implementation, may even have the opposite effect.

6.3 Limitations of the Evaluation Setup

Our approach to assessing the quality of compressed segments was to truncate each baseline segment to the length of a model's hypothesis. However, this BLEU baseline score was in all cases significantly higher than all the results achieved within our experiments. In addition to the comparison with several references that are partly responsible for the extremely high BLEU scores, we suspect that this is because the annotators did not rephrase the source sentences, but mostly just deleted superfluous words, resulting in high ngram matches when calculating BLEU. This shows that evaluating sentence compression remains a difficult task. We propose to evaluate future experiments using, inter alia, a combination of text simplification metrics and back-translations to simulate a monolingual evaluation.

7 Conclusion

In this thesis, we have investigated different approaches to integrate the sentence compression into machine translation. Firstly, we developed approaches to preprocess training data in a way that a model could gain length awareness and learn to perform sentence compression. We experimented with different kinds of tags, used factored machine translation and also included monolingual compression data into our models. We also implemented a new length penalty that does not require any training or finetuning, but is solely based on modifying the beam search algorithm.

As for our first research question, all of our data-driven approaches were successful in teaching the trained models length awareness and the ability to generate compressed translations, albeit with varying degrees of success. The first approach, which was solely based on annotating existing parallel data based on the length of each segment, showed surprisingly good ability to generate shorter translations. However, it remains unclear whether these translations are just short or indeed compressions. The additional information provided by the length factors only resulted in a suboptimal compression result, which is why this approach fell short of expectations. However, there are indications that, if the expected target length is known or anticipated correctly, regular translation quality may benefit from length factors. Unsurprisingly, the best data-driven approach was the one that was trained on back-translated compression data, with the added benefit of being trained on more training segments than the other approaches. We also found that in all of our models, learning the ability to compress sentences did not impair the quality of uncompressed translations, but rather improved them.

With regards to our second research question, the limited decoding approach was unsuccessful: The results showed that significant compression was not possible, while the translation quality deteriorated significantly. Accordingly, the the introduction of a new length penalty did have an impact on the decoding algorithm, but not to the extent hoped for.

In conclusion, it has been shown that integrating (rare) compression data is currently the best way to achieve multilingual sentence compression. Although a purely

decoding-based approach would be desirable, it seems not possible to reduce the translation length without a significant loss of quality without length awareness of the model. On the positive side, we showed that regular parallel data can also be used to teach a model to some extent the ability to shorten translations through data annotation. Future work will have to explore the possibilities of leveraging this abundant data source.

References

M. Angrosh, T. Nomoto, and A. Siddharthan. Lexico-syntactic text simplification and compression with typed dependencies. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1996–2006, Dublin, Ireland, Aug. 2014. Dublin City University and Association for Computational Linguistics. URL https://aclanthology.org/C14-1188.

I. Caswell, C. Chelba, and D. Grangier. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5206. URL https://aclanthology.org/W19-5206.

M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi. MuST-C: a Multilingual Speech Translation Corpus. 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 2012–2017, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1202. URL https://aclanthology.org/N19-1202.

G. Dinu, P. Mathur, M. Federico, and Y. Al-Onaizan. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1294. URL https://aclanthology.org/P19-1294.

S. Edunov, M. Ott, M. Auli, and D. Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1045. URL https://aclanthology.org/D18-1045.

T. Etchegoyhen and H. Gete. To case or not to case: Evaluating casing methods for neural machine translation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3752–3760, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.463.

K. Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 322–330, Beijing, China, Aug. 2010. Coling 2010 Organizing Committee. URL https://aclanthology.org/C10-1037.

K. Filippova and Y. Altun. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1155.

K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals. Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1042. URL https://aclanthology.org/D15-1042.

F. Hieber, M. Denkowski, T. Domhan, B. D. Barros, C. D. Ye, X. Niu, C. Hoang, K. Tran, B. Hsu, M. Nadejde, S. Lakew, P. Mathur, A. Currey, and M. Federico. Sockeye 3: Fast neural machine translation with pytorch, 2022. URL https://arxiv.org/abs/2207.05851.

C. Hokamp and Q. Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1141. URL https://aclanthology.org/P17-1141.

M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://aclanthology.org/Q17-1024.

J. Jon, J. P. Aires, D. Varis, and O. Bojar. End-to-end lexically constrained machine translation for morphologically rich languages. 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 4019–4033, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.311. URL https://aclanthology.org/2021.acl-long.311.

Y. Kikuchi, G. Neubig, R. Sasano, H. Takamura, and M. Okumura. 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, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1140. URL https://aclanthology.org/D16-1140.

P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit 2005*, pages 79–86, 2005. URL http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/europarl-mtsummit05.pdf.

J. Kreutzer, J. Bastings, and S. Riezler. Joey NMT: A minimalist NMT toolkit for novices. 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): System Demonstrations*, pages 109–114, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-3019. URL https://aclanthology.org/D19-3019.

S. M. Lakew, M. Di Gangi, and M. Federico. Controlling the output length of neural machine translation. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong, Nov. 2-3 2019. Association for Computational Linguistics. URL https://aclanthology.org/2019.iwslt-1.31.

Z. Li, R. Wang, K. Chen, M. Utiyama, E. Sumita, Z. Zhang, and H. Zhao. Explicit sentence compression for neural machine translation. *CoRR*, abs/1912.11980, 2019. URL http://arxiv.org/abs/1912.11980.

J. Mallinson, R. Sennrich, and M. Lapata. Sentence compression for arbitrary languages via multilingual pivoting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2453–2464, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1267. URL https://aclanthology.org/D18-1267.

M. Müller and R. Sennrich. Understanding the properties of minimum Bayes risk decoding in neural 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 259–272, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.22. URL https://aclanthology.org/2021.acl-long.22.

K. Murray and D. Chiang. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6322. URL https://aclanthology.org/W18-6322.

J. Niehues. Machine translation with unsupervised length-constraints. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 21–35, Virtual, Oct. 2020. Association for Machine Translation in the Americas. URL https://aclanthology.org/2020.amta-research.3.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 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, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

E. Pitler. Methods for sentence compression, May 2010. URL https://repository.upenn.edu/cis_reports/929/.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319.

S. Roy. Generating summaries using sentence compression and statistical measures, Oct 2020. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3681279.

R. Sennrich and B. Haddow. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, Aug. 2016.

Association for Computational Linguistics. doi: 10.18653/v1/W16-2209. URL https://aclanthology.org/W16-2209.

R. Sennrich, B. Haddow, and A. Birch. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1005. URL https://aclanthology.org/N16-1005.

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, Aug. 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009.

R. Sennrich, B. Haddow, and A. Birch. 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, Aug. 2016c. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL http://arxiv.org/abs/1409.3215.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL https://arxiv.org/abs/1609.08144.

H. Yamagishi, S. Kanouchi, T. Sato, and M. Komachi. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee. URL `https://aclanthology.org/W16-4620`.

N. Yu, J. Zhang, M. Huang, and X. Zhu. An operation network for abstractive sentence compression. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1065–1076, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics. URL `https://aclanthology.org/C18-1091`.