Journal of Artificial Intelligence Research 69 (2020) 343-418 Submitted 03/2020; published 10/2020

Neural Machine Translation: A Review

Felix Stahlberg FS439@CANTAB.AC.UK
University of Cambridge, Engineering Department, Trumpington Street
Cambridge CB2 1PZ, United Kingdom

Abstract

The field of machine translation (MT), the automatic translation of written text from one natural language into another, has experienced a major paradigm shift in recent years. Statistical MT, which mainly relies on various count-based models and which used to dominate MT research for decades, has largely been superseded by neural machine translation (NMT), which tackles translation with a single neural network. In this work we will trace back the origins of modern NMT architectures to word and sentence embeddings and earlier examples of the encoder-decoder network family. We will conclude with a short survey of more recent trends in the field.

1. Introduction

Various fields in the area of natural language processing (NLP) have been boosted by the rediscovery of neural networks (see Goldberg, 2016 for an overview). However, for a long time, the integration of neural nets into machine translation (MT) systems was rather shallow. Early attempts used feedforward neural language models (Bengio et al., 2003, 2006) for the target language to rerank translation lattices (Schwenk et al., 2006). The first neural models which also took the source language into account extended this idea by using the same model with bilingual tuples instead of target language words (Zamora-Martinez et al., 2010), scoring phrase pairs directly with a feedforward net (Schwenk, 2012), or adding a source context window to the neural language model (Le et al., 2012; Devlin et al., 2014). Kalchbrenner and Blunsom (2013) and Cho et al. (2014b) introduced recurrent networks for translation modelling. All those approaches applied neural networks as components in a traditional statistical machine translation system. Therefore, they retained the log-linear model combination and only exchanged parts in the traditional architecture.

Neural machine translation (NMT) has overcome this separation by using a single large neural net that directly transforms the source sentence into the target sentence (Cho et al., 2014a; Sutskever et al., 2014; Bahdanau et al., 2015). The advent of NMT certainly marks one of the major milestones in the history of MT, and has led to a radical and sudden departure of mainstream research from many previous research lines. This is perhaps best reflected by the explosion of scientific publications related to NMT in the past few years[1] (Fig. 1), and the large number of publicly available NMT toolkits (Tab. 1). NMT has already been widely adopted in industry (Wu et al., 2016; Crego et al., 2016; Schmidt & Marg, 2018; Levin et al., 2017) and is deployed in production systems by Google, Microsoft, Facebook, Amazon, SDL, Yandex, and many more. This article will introduce the basic concepts of NMT, and will give an overview of current research in the field.

1. Example Google Scholar search: `https://scholar.google.com/scholar?q=%22neural+machine+translation%22&as_ylo=2017&as_yhi=2017`

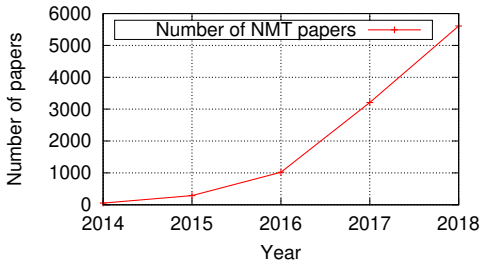


Figure 1: Number of papers mentioning "neural machine translation" per year according Google Scholar.

Name	Citation	Framework	GitHub Stars
Tensor2Tensor	Vaswani et al. (2018)	TensorFlow	
TensorFlow/NMT	-	TensorFlow	
Fairseq	Ott et al. (2019)	PyTorch	
OpenNMT-py	Klein et al. (2017)	Lua, (Py)Torch, TF	
Sockeye	Hieber et al. (2017)	MXNet	
OpenSeq2Seq	Kuchaiev et al. (2018)	TensorFlow	
Nematus	Sennrich et al. (2017b)	TensorFlow, Theano	
PyTorch/Translate	-	PyTorch	
Marian	Junczys-Dowmunt et al. (2016a)	C++	
NMT-Keras	Álvaro Peris and Casacuberta (2018)	TensorFlow, Theano	
Neural Monkey	Helcl and Libovický (2017)	TensorFlow	
THUMT	Zhang et al. (2017)	TensorFlow, Theano	
Eske/Seq2Seq	-	TensorFlow	
XNMT	Neubig et al. (2018)	DyNet	
NJUNMT	-	PyTorch, TensorFlow	
Transformer-DyNet	-	DyNet	
SGNMT	Stahlberg et al. (2017, 2018)	TensorFlow, Theano	
CythonMT	Wang et al. (2018)	C++	
Neutron	Xu and Liu (2019)	PyTorch	

Table 1: NMT tools that have been updated in the past year (as of 2019). GitHub stars indicate the popularity of tools on GitHub.

2. Nomenclature

We will denote the source sentence of length I as $\mathbf{x}$. We use the subscript i to index tokens in the source sentence. We refer to the source language vocabulary as Σ_{src}.

$$\mathbf{x} = x_1^I = (x_1, \ldots, x_I) \in \Sigma_{src}^I \tag{1}$$

The translation of source sentence $\mathbf{x}$ into the target language is denoted as $\mathbf{y}$. We use an analogous nomenclature on the target side.

$$\mathbf{y} = y_1^J = (y_1, \ldots, y_J) \in \Sigma_{trg}^J \tag{2}$$

In case we deal with only one language we drop the subscript src/trg. For convenience we represent tokens as indices in a list of subwords or word surface forms. Therefore, Σ_{src}

and Σ_{trg} are the first n natural numbers (i.e. $\Sigma = \{n' \in \mathbb{N} | n' \leq n\}$ where $n = |\Sigma|$ is the vocabulary size). Additionally, we use the projection function π_k which maps a tuple or vector to its k-th entry:

$$\pi_k(z_1, \ldots, z_k, \ldots, z_n) = z_k. \tag{3}$$

For a matrix $A \in \mathbb{R}^{m \times n}$ we denote the element in the p-th row and the q-th column as $A_{p,q}$, the p-th row vector as $A_{p,:} \in \mathbb{R}^n$ and the q-th column vector as $A_{:,q} \in \mathbb{R}^m$. For a series of m n-dimensional vectors $a_p \in \mathbb{R}^n$ ($p \in [1, m]$) we denote the $m \times n$ matrix which results from stacking the vectors horizontally as $(a_p)_{p=1:m}$ as illustrated with the following tautology:

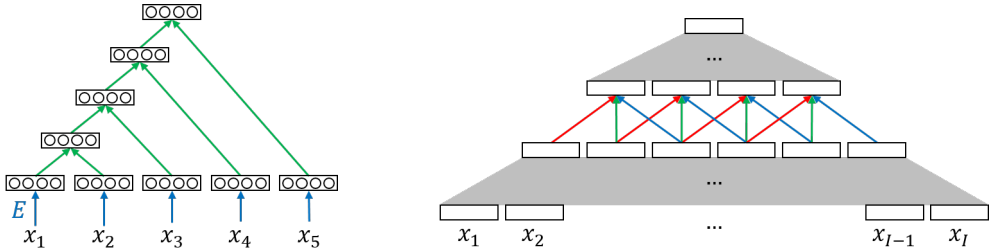
$$A = (A_{p,:})_{p=1:m} = ((A_{:,q})_{q=1:n})^T. \tag{4}$$

3. Word Embeddings

Representing words or phrases as continuous vectors is arguably one of the keys in connectionist models for NLP. One of the early successful applications of continuous space word representations were language models (Bellegarda, 1997; Bengio et al., 2003). The key idea is to represent a word $x \in \Sigma$ as a d-dimensional vector of real numbers. The size d of the embedding layer is normally chosen to be much smaller than the vocabulary size ($d \ll |\Sigma|$). The mapping from the word to its distributed representation can be represented by an embedding matrix $E \in \mathbb{R}^{d \times |\Sigma|}$ (Collobert & Weston, 2008). The x^{th} column of E (denoted as $E_{:,x}$) holds the d-dimensional representation for the word x.

Learned continuous word representations have the potential of capturing morphological, syntactic and semantic similarity across words (Collobert & Weston, 2008). In neural machine translation, embedding matrices are usually trained jointly with the rest of the network using backpropagation (Rumelhart et al., 1988) and a gradient based optimizer such as stochastic gradient descent. In other areas of NLP, pre-trained word embeddings trained on unlabelled text have become ubiquitous (Collobert et al., 2011). Methods for training word embeddings on raw text often take the context into account in which the word occurs frequently (Pennington et al., 2014; Mikolov et al., 2013a), or use cross-lingual information to improve embeddings (Mikolov et al., 2013b; Upadhyay et al., 2016).

A newly emerging type of *contextualized* word embeddings (Peters et al., 2017; McCann et al., 2017) is gaining popularity in various fields of NLP. Contextualized representations do not only depend on the word itself but on the entire input sentence. Thus, they cannot be described by a single embedding matrix but are usually generated by neural sequence models which have been trained under a language model objective. Most approaches either use LSTM or Transformer architectures but differ in the way these architectures are used to compute the word representations (Peters et al., 2017, 2018; Radford et al., 2018; Devlin et al., 2019). Contextualized word embeddings have advanced the state-of-the-art in several NLP benchmarks (Peters et al., 2018; Bowman et al., 2018; Devlin et al., 2019). Goldberg (2019) showed that contextualized embeddings are remarkably sensitive to syntax. Choi et al. (2017) reported gains from contextualizing word embeddings in NMT using a bag of words.



(a) Recursive autoencoder following Socher et al. (2011).

(b) The convolutional sentence model (Kalchbrenner & Blunsom, 2013).

Figure 2: Phrase and sentence embedding architectures. The color coding indicates weight sharing.

4. Phrase and Sentence Embeddings

For various NLP tasks such as sentiment analysis or MT it is desirable to embed whole phrases or sentences instead of single words. For example, a distributed representation of the source sentence $\mathbf{x}$ could be used as conditional for the distribution over the target sentences $P(\mathbf{y}|\mathbf{x})$. Early approaches to phrase embedding were based on recurrent autoencoders (Pollack, 1990; Socher et al., 2011). To represent a phrase $\mathbf{x} \in \Sigma^I$ as a d-dimensional vector, Socher et al. (2011) first trained a word embedding matrix $E \in \mathbb{R}^{d \times |\Sigma|}$. Then, they recursively applied an autoencoder network which finds d-dimensional representations for $2d$-dimensional inputs, where the input is the concatenation of two child representations. The child representations are either word embeddings or representations calculated by the same autoencoder from two different parents. The order in which representations are merged is determined by a binary tree over $\mathbf{x}$ which can be constructed greedily (Socher et al., 2011) or derived from an Inversion Transduction Grammar (Wu, 1997; Li et al., 2013). Fig. 2a shows an example of a recurrent autoencoder embedding a phrase with five words into a four dimensional space. One of the disadvantages of recurrent autoencoders is that the word and sentence embeddings need to have the same dimensionality. This restriction is not very critical in sentiment analysis because the sentence representation is only used to extract the sentiment of the writer (Socher et al., 2011). In MT, however, the sentence representations need to convey enough information to condition the target sentence distribution on it, and thus should be higher dimensional than the word embeddings.

Kalchbrenner and Blunsom (2013) used convolution to find vector representations of phrases or sentences and thus avoided the dimensionality issue of recurrent autoencoders. As shown in Fig. 2b, their model yields n-gram representations at each convolution level, with n increasing with depth. The top level can be used as a representation for the whole sentence. Other notable examples of using convolution for sentence representations include (Kalchbrenner et al., 2014; Kim, 2014; Mou et al., 2016; dos Santos & Gatti, 2014; Er et al., 2016). However, the convolution operations in these models lose information about the exact word order, and are thus more suitable for sentiment analysis than for tasks like machine translation.[2] A recent line of work uses self-attention (Sec. 6.5) rather than convolution to find sentence representations (Shen et al., 2018a; Wu et al., 2018; Zhang

2. This is not to be confused with convolutional *translation* models which will be reviewed in Sec. 6.4.

et al., 2018a). Another idea explored by Yu et al. (2018) is to resort to (recursive) relation networks (Santoro et al., 2017; Palm et al., 2018) which repeatedly aggregate pairwise relations between words in the sentence. Recurrent architectures are also commonly used for sentence representation. It has been noted that even random RNNs without any training can work surprisingly well for several NLP tasks (Conneau et al., 2017, 2018; Wieting & Kiela, 2019).

5. Encoder-Decoder Networks with Fixed Length Sentence Encodings

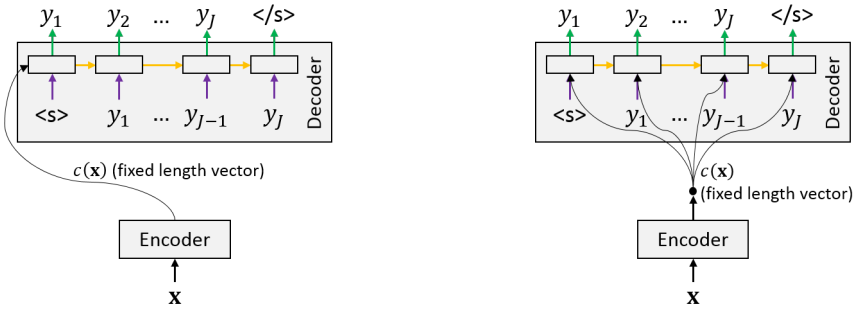
Kalchbrenner and Blunsom (2013) were the first who conditioned the target sentence distribution on a distributed fixed-length representation of the source sentence. Their recurrent continuous translation models (RCTM) I and II followed the family of so-called encoder-decoder networks (Neco & Forcada, 1997) which is the current prevailing architecture for NMT. Encoder-decoder networks are subdivided into an encoder network which computes a representation of the source sentence, and a decoder network which generates the target sentence from that representation. As introduced in Sec. 2 we denote the source sentence as $\mathbf{x} = x_1^I$ and the target sentence as $\mathbf{y} = y_1^J$. Most existing NMT models are auto-regressive and thus define a probability distribution over the target sentences $P(\mathbf{y}|\mathbf{x})$ by factorizing it into conditionals:

$$P(\mathbf{y}|\mathbf{x}) \overset{\text{Chain rule}}{=} \prod_{j=1}^{J} P(y_j|y_1^{j-1}, \mathbf{x}). \tag{5}$$

Different encoder-decoder architectures differ vastly in how they model the distribution $P(y_j|y_1^{j-1}, \mathbf{x})$. We will first discuss encoder-decoder networks in which the encoder represents the source sentence as a fixed-length vector $c(\mathbf{x})$ like the methods in Sec. 4. The conditionals $P(y_j|y_1^{j-1}, \mathbf{x})$ are modelled as:

$$P(y_j|y_1^{j-1}, \mathbf{x}) = g(y_j|s_j, y_{j-1}, c(\mathbf{x})) \tag{6}$$

where s_j is the hidden state of a recurrent neural (decoder) network (RNN). We will formally introduce s_j in Sec. 6.3. Gated activation functions such as the long short-term memory (Hochreiter & Schmidhuber, 1997, LSTM) or the gated recurrent unit (Cho et al., 2014b, GRU) are commonly used to alleviate the vanishing gradient problem (Hochreiter et al., 2001) which makes it difficult to train RNNs to capture long-range dependencies. Deep architectures with stacked LSTM cells were used by Sutskever et al. (2014). The encoder can be a convolutional network as in the RCTM I (Kalchbrenner & Blunsom, 2013), an LSTM network (Sutskever et al., 2014), or a GRU network (Cho et al., 2014b). $g(\cdot)$ is a feedforward network with a softmax layer at the end which takes as input the decoder state s_j and an embedding of the previous target token y_{j-1}. In addition, $g(\cdot)$ may also take the source sentence encoding $c(\mathbf{x})$ as input to condition on the source sentence (Kalchbrenner & Blunsom, 2013; Cho et al., 2014b). Alternatively, $c(\mathbf{x})$ is just used to initialize the decoder state s_1 (Sutskever et al., 2014; Bahdanau et al., 2015). Fig. 3 contrasts both methods. Intuitively, once the source sentence has been encoded, the decoder starts generating the first target sentence symbol y_1 which is then fed back to the decoder network for producing the second symbol y_2. The algorithm terminates when the network produces the end-of-sentence symbol </s>. Sec. 7 explains more formally what we mean by the network "generating" a



(a) Source sentence is used to initialize the decoder state.

(b) Source sentence is fed to the decoder at each time step.

Figure 3: Encoder-decoder architectures with fixed-length sentence encodings. The color coding indicates weight sharing.

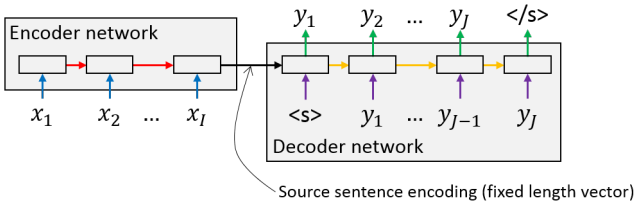


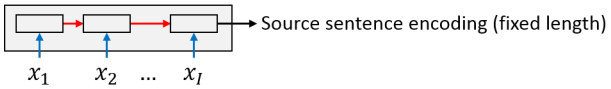
Figure 4: The encoder-decoder architecture of Sutskever et al. (2014). The color coding indicates weight sharing.

symbol y_j and sheds more light on the aspect of decoding in NMT. Fig. 4 shows the complete architecture of Sutskever et al. (2014) who presented one of the first working standalone NMT systems that did not rely on any SMT baseline. One of the reasons why this paper was groundbreaking is the simplicity of the architecture, which stands in stark contrast to traditional SMT systems that used a very large number of highly engineered features.

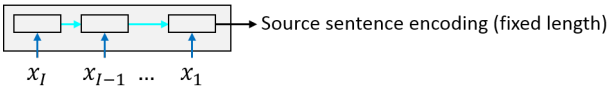
Different ways of providing the source sentence to the encoder network have been explored in the past. Cho et al. (2014b) fed the tokens to the encoder in the natural order they appear in the source sentence (cf. Fig. 5a). Sutskever et al. (2014) reported gains from simply feeding the sequence in reversed order (cf. Fig. 5b). They argue that these improvements might be "caused by the introduction of many short term dependencies to the dataset" (Sutskever et al., 2014). Bidirectional RNNs (Schuster & Paliwal, 1997, BiRNN) are able to capture both directions (cf. Fig. 5c) and are often used in attentional NMT (Bahdanau et al., 2015).

6. Attentional Encoder-Decoder Networks

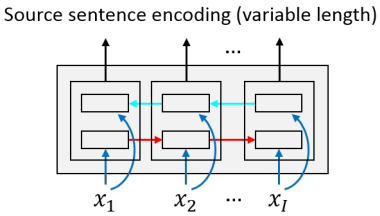
One problem of early NMT models which is not fully solved yet (see Sec. 8.1) is that they often produced poor translations for long sentences (Sountsov & Sarawagi, 2016). Cho et al. (2014a) suggested that this weakness is due to the fixed-length source sentence encoding. Sentences with varying length convey different amounts of information. Therefore, despite



(a) Unidirectional encoder used by Cho et al. (2014b).



(b) Reversed unidirectional encoder from Sutskever et al. (2014).



(c) Bidirectional encoder used by Bahdanau et al. (2015).

Figure 5: Encoder architectures. The color coding indicates weight sharing.

being appropriate for short sentences, a fixed-length vector "does not have enough capacity to encode a long sentence with complicated structure and meaning" (Cho et al., 2014a). Pouget-Abadie et al. (2014) tried to mitigate this problem by chopping the source sentence into short clauses. They composed the target sentence by concatenating the separately translated clauses. However, this approach does not cope well with long-distance reorderings as word reorderings are only possible within a clause.

6.1 Attention

Bahdanau et al. (2015) introduced the concept of *attention* to avoid having a fixed-length source sentence representation. Their model does not use a constant context vector $c(\mathbf{x})$ any more which encodes the whole source sentence. By contrast, the attentional decoder can place its attention on parts of the source sentence which are useful for producing the next token. The constant context vector $c(\mathbf{x})$ is thus replaced by a series of context vectors $c_j(\mathbf{x})$; one for each time step j.[3]

We will first introduce attention as a general concept before describing the architecture of Bahdanau et al. (2015) in detail in Sec. 6.3. We follow the terminology of Vaswani et al. (2017) and describe attention as mapping n query vectors to n output vectors via a mapping table (or a *memory*) of m key-value pairs. We make the simplifying assumption that all vectors have the same dimension d so that we can stack the vectors into matrices $Q \in \mathbb{R}^{n \times d}$, $K \in \mathbb{R}^{m \times d}$, and $V \in \mathbb{R}^{m \times d}$. Intuitively, for each query vector we compute an

3. We refer to j as 'time step' due to the sequential structure of autoregressive models and the left-to-right order of NMT decoding. We note, however, that j does not specify a point in time in the usual sense but rather the position in the target sentence.

Name	Scoring function	Citation
Additive	$\text{score}(Q,K)_{p,q} = v^\intercal \tanh(WQ_{p,:} + UK_{q,:})$	Bahdanau et al. (2015)
Dot-product	$\text{score}(Q,K) = QK^\intercal$	Luong et al. (2015a)
Scaled dot-product	$\text{score}(Q,K) = QK^\intercal d^{-0.5}$	Vaswani et al. (2017)

Table 2: Common attention scoring functions. $v \in \mathbb{R}^{d_{\text{att}}}$, $W \in \mathbb{R}^{d_{\text{att}} \times d}$, and $U \in \mathbb{R}^{d_{\text{att}} \times d}$ in additive attention are trainable parameters with d_{att} being the dimensionality of the attention layer.

output vector as a weighted sum of the value vectors. The weights are determined by a similarity score between the query vector and the keys (cf. Vaswani et al., 2017, Eq. 1):

$$\underbrace{\text{Attention}(K,V,Q)}_{n \times d} = \text{Softmax}(\underbrace{\text{score}(Q,K)}_{n \times m}) \underbrace{V}_{m \times d}. \tag{7}$$

The output of $\text{score}(Q,K)$ is an $n \times m$ matrix of similarity scores. The softmax function normalizes over the columns of that matrix so that the weights for each query vector sum up to one. A straightforward choice for $\text{score}(\cdot)$ proposed by Luong et al. (2015a) is the dot product (i.e. $\text{score}(Q,K) = QK^\intercal$). The most common scoring functions are summarized in Tab. 2.

A common way to use attention in NMT is at the interface between encoder and decoder. Bahdanau et al. (2015), Luong et al. (2015a) used the hidden decoder states s_j as query vectors. Both the key and value vectors are derived from the hidden states h_i of a recursive encoder.[4] Formally, this means that $Q = s_j$ are the query vectors , $n = J$ is the target sentence length, $K = V = h_i$ are the key and value vectors, and $m = I$ is the source sentence length.[5] The outputs of the attention layer are used as time-dependent context vectors $c_j(\mathbf{x})$. In other words, rather than using a fixed-length sentence encoding $c(\mathbf{x})$ as in Sec. 5, at each time step j we query a memory in which entries store (context-sensitive) representations of the source words. In this setup it is possible to derive an attention matrix $A \in \mathbb{R}^{J \times I}$ to visualize the learned relations between words in the source sentence and words in the target sentence:

$$A := \text{Softmax}(\text{score}((s_j)_{j=1:J}, (h_i)_{i=1:I})). \tag{8}$$

Fig. 6 shows an example of A from an English-German NMT system with additive attention. The attention matrix captures cross-lingual word relationships such as "is" → "ist" or "great" → "großer". The system has learned that the English source word "is" is relevant for generating the German target word "ist" and thus emits a high attention weight for this pair. Consequently, the context vector $c_j(\mathbf{x})$ at time step $j = 3$ mainly represents the source word "is" ($c_3(\mathbf{x}) \approx h_2$). This is particularly significant as the system was not explicitly trained to align words but to optimize translation performance. As an alternative to this alignment perspective, attention also has a probabilistic interpretation as the attention matrix A contains valid probability distributions which are used to take the expectation over the values.

4. s_j and h_i are defined in Sec. 5 and Sec. 6.3.
5. An exception is the model of Mino et al. (2017) that splits h_i into two parts and uses the first part as key and the second as value.

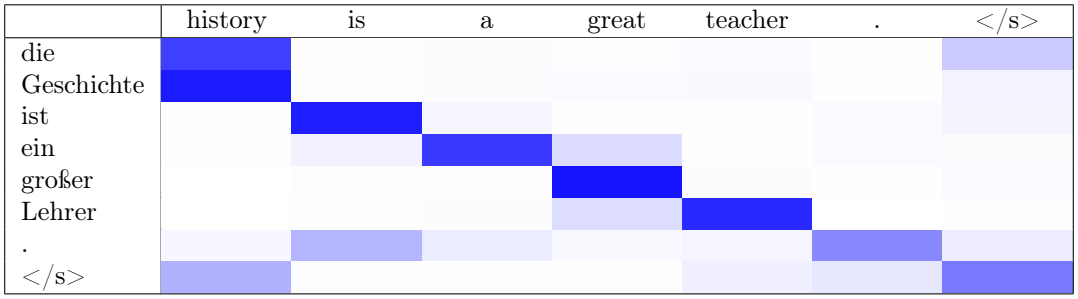


Figure 6: Attention weight matrix A for the translation from the English sentence "history is a great teacher ." to the German sentence "die Geschichte ist ein großer Lehrer .". Dark shades of blue indicate high attention weights.

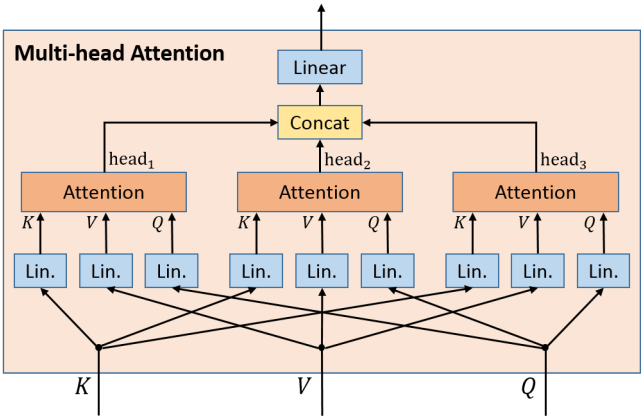


Figure 7: Multi-head attention with three attention heads.

An important generalization of attention is *multi-head* attention proposed by Vaswani et al. (2017). The idea is to perform H attention operations instead of a single one where H is the number of attention heads (usually $H = 8$). The query, key, and value vectors for the attention heads are linear transforms of Q, K, and V. The output of multi-head attention is the concatenation of the outputs of each attention head. The dimensionality of the attention heads is usually divided by H to avoid increasing the number of parameters. Formally, it can be described as follows (Vaswani et al., 2017):

$$\text{MultiHeadAttention}(K, V, Q) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)W^O \tag{9}$$

with weight matrix $W^O \in \mathbb{R}^{d \times d}$ where

$$\text{head}_h = \text{Attention}(KW_h^K, VW_h^V, QW_h^Q) \tag{10}$$

with weight matrices $W_h^K, W_h^V, W_h^Q \in \mathbb{R}^{d \times \frac{d}{H}}$ for $h \in [1, H]$. Fig. 7 shows a multi-head attention module with three heads. Multi-head attention can be viewed as multiple networks running in parallel with different views on the key-value set (e.g. to capture varying linguistic phenomena) and map them to different subspaces of the output representation.[6] However,

6. We thank one of the anonymous reviewers for making this point.

the	first	cold	shower	<pad>	<pad>
even	the	monkey	seems	to	want
a	little	coat	of	straw	<pad>

Figure 8: A tensor containing a batch of three source sentences of different lengths ("the first cold shower", "even the monkey seems to want", "a little coat of straw" – a haiku by Basho (Basho & Reichhold, 2013)). Short sentences are padded with <pad>. The training loss and attention masks are visualized with green (enabled) and red (disabled) background.

it is not obvious anymore how to derive a single attention weight matrix A like shown in Fig. 6. Therefore, models using multi-head attention tend to be more difficult to interpret.

The concept of attention is no longer just a technique to improve the translation of long sentences. Since its introduction by Bahdanau et al. (2015) it has become a vital part of various NMT architectures, culminating in the Transformer architecture (Sec. 6.5) which is entirely attention-based. Attention has also been proven effective for, inter alia, object recognition (Larochelle & Hinton, 2010; Ba et al., 2014; Mnih et al., 2014), image caption generation (Xu et al., 2015), video description (Yao et al., 2015), speech recognition (Chorowski et al., 2014; Chan et al., 2016), cross-lingual word-to-phone alignment (Duong et al., 2016), bioinformatics (Sønderby et al., 2015), text summarization (Rush et al., 2015), text normalization (Sproat & Jaitly, 2016), grammatical error correction (Yuan & Briscoe, 2016), question answering (Hermann et al., 2015; Yang et al., 2016; Sukhbaatar et al., 2015), natural language understanding and inference (Dong & Lapata, 2016; Shen et al., 2018a; Im & Cho, 2017; Liu et al., 2016), uncertainty detection (Adel & Schütze, 2017), photo optical character recognition (Lee & Osindero, 2016), and natural language conversation (Shang et al., 2015).

6.2 Attention Masks and Padding

NMT usually groups sentences into batches to make more efficient use of the available hardware and to reduce noise in gradient estimation. However, the central data structure for many machine learning frameworks (Bastien et al., 2012; Abadi et al., 2016) are *tensors* – multi-dimensional arrays with fixed dimensionality. Re-arranging source sentences as tensor often results in some unused space as the sentences may vary in length. In practice, shorter sentences are filled up with a special padding symbol <pad> to match the length of the longest sentence in the batch (Fig. 8). Most implementations work with masks to avoid taking padded positions into account when computing the training loss. Attention layers also have to be restricted to non-padding symbols which is also usually realized by multiplying the attention weights by a mask that sets the attention weights for padding symbols to zero. Sentences of similar lengths are often grouped into batches to minimize padding and thereby increase the efficiency.

6.3 Recurrent Neural Machine Translation

This section contains a complete formal description of the RNNsearch architecture of Bahdanau et al. (2015) which was the first NMT model using attention. Recall that NMT uses the chain rule to decompose the probability $P(\mathbf{y}|\mathbf{x})$ of a target sentence $\mathbf{y} = y_1^J$ given

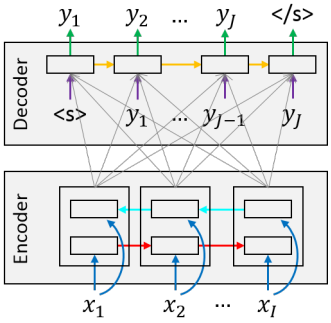


Figure 9: The RNNsearch model following Bahdanau et al. (2015). The color coding indicates weight sharing. Gray arrows represent attention.

a source sentence $\mathbf{x} = x_1^I$ into left-to-right conditionals (Eq. 5). RNNsearch models the conditionals as follows (Bahdanau et al., 2015, Eq. 2,4):

$$P(\mathbf{y}|\mathbf{x}) \overset{\text{Eq. 5}}{=} \prod_{j=1}^{J} P(y_j|y_1^{j-1}, \mathbf{x}) = \prod_{j=1}^{J} g(y_j|y_{j-1}, s_j, c_j(\mathbf{x})). \tag{11}$$

Similarly to Eq. 6, the function $g(\cdot)$ encapsulates the decoder network which computes the distribution for the next target token y_j given the last produced token y_{j-1}, the RNN decoder state $s_j \in \mathbb{R}^n$, and the context vector $c_j(\mathbf{x}) \in \mathbb{R}^m$. The sizes of the encoder and decoder hidden layers are denoted with m and n. The context vector $c_j(\mathbf{x})$ is a distributed representation of the relevant parts of the source sentence. In NMT without attention (Sec. 5), the context vector is constant and thus needs to encode the whole source sentence. Adding an attention mechanism results in different context vectors for each target sentence position j. This effectively addresses issues in NMT due to the limited capacity of a fixed context vector as illustrated in Fig. 9.

As outlined in Sec. 6.1, the context vectors $c_j(\mathbf{x})$ are weighted sums of source sentence *annotations* $\mathbf{h} = (h_1, \dots, h_I)$. The annotations are produced by the encoder network. In other words, the encoder converts the input sequence $\mathbf{x}$ to a sequence of annotations $\mathbf{h}$ of the same length. Each annotation $h_i \in \mathbb{R}^m$ encodes information about the entire source sentence $\mathbf{x}$ "with a strong focus on the parts surrounding the i-th word of the input sequence" (Bahdanau et al., 2015, Sec. 3.1). RNNsearch uses a bidirectional RNN (Fig. 5c) to generate the annotations. A BiRNN consists of two independent RNNs. The forward RNN $\overrightarrow{f}$ reads $\mathbf{x}$ in the original order (from x_1 to x_I). The backward RNN $\overleftarrow{f}$ consumes $\mathbf{x}$ in reversed order (from x_I to x_1):

$$\overrightarrow{h}_i = \overrightarrow{f}(x_i, \overrightarrow{h}_{i-1}) \tag{12}$$

$$\overleftarrow{h}_i = \overleftarrow{f}(x_i, \overleftarrow{h}_{i+1}). \tag{13}$$

The RNNs $\overrightarrow{f}(\cdot)$ and $\overleftarrow{f}(\cdot)$ are usually LSTM or GRU cells. The annotation h_i is the concatenation of the hidden states $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ (Bahdanau et al., 2015, Sec. 3.2):

$$h_i = [\overrightarrow{h}_i^\intercal; \overleftarrow{h}_i^\intercal]^\intercal. \tag{14}$$

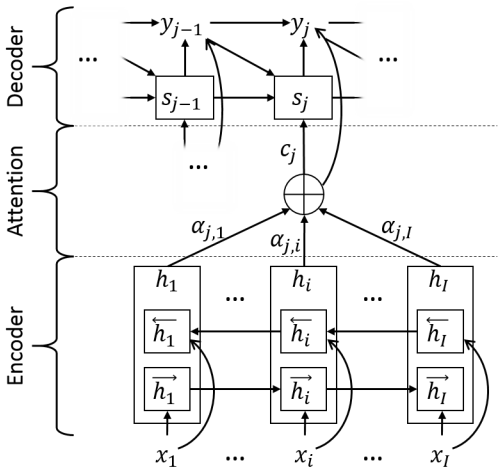


Figure 10: Illustration of the attention mechanism in RNNsearch (Bahdanau et al., 2015).

The context vectors $c_j(\mathbf{x}) \in \mathbb{R}^m$ are computed from the annotations as weighted sums with weights $\alpha_j \in [0,1]^I$ (Bahdanau et al., 2015, Eq. 5):

$$c_j(\mathbf{x}) = \sum_{i=1}^{I} \alpha_{j,i} h_i. \tag{15}$$

The weights are determined by the alignment model $a(\cdot)$:

$$\alpha_{j,i} = \frac{1}{Z} \exp(a(s_{j-1}, h_i)) \text{ with } Z = \sum_{k=1}^{I} \exp(a(s_{j-1}, h_k)) \tag{16}$$

where $a(s_{j-1}, h_i)$ is a feedforward neural network which estimates the importance of annotation h_i for producing the j-th target token given the current decoder state $s_{j-1} \in \mathbb{R}^n$. In the terminology of Sec. 6.1, h_i represent the keys and values, s_j are the queries, and $a(\cdot)$ is the attention scoring function.

The function $g(\cdot)$ in Eq. 11 not only takes the previous target token y_{j-1} and the context vector c_j but also the decoder hidden state s_j.

$$s_j = f(s_{j-1}, y_{j-1}, c_j) \tag{17}$$

where $f(\cdot)$ is modelled by a GRU or LSTM cell. The function $g(\cdot)$ is defined as follows.

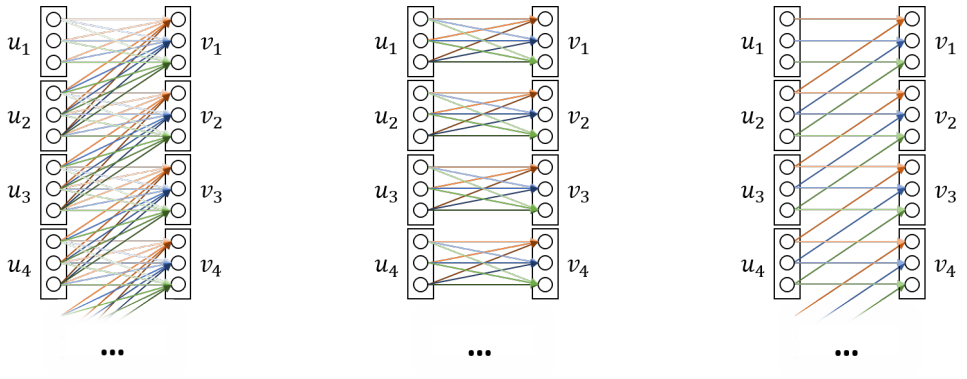
$$g(y_j | y_{j-1}, s_j, c_j) \propto \exp(W_o \max(t_j, u_j)) \tag{18}$$

with

$$t_j = T_s s_j + T_y E y_{j-1} + T_c c_j \tag{19}$$

$$u_j = U_s s_j + U_y E y_{j-1} + U_c c_j \tag{20}$$

where $\max(\cdot)$ is the *element-wise* maximum, and $W_o \in \mathbb{R}^{|\Sigma_{trg}| \times l}$, $T_s, U_s \in \mathbb{R}^{l \times n}$, $T_y, U_y \in \mathbb{R}^{l \times k}$, $E \in \mathbb{R}^{k \times |\Sigma_{trg}|}$, $T_c, U_c \in \mathbb{R}^{l \times m}$ are weight matrices. The definition of $g(\cdot)$ can be seen



(a) Standard convolution. (b) Pointwise convolution. (c) Depthwise convolution.

Figure 11: Types of 1D-convolution used in NMT. The color coding indicates weight sharing.

as connecting the output of the recurrent layer, a k-dimensional embedding of the previous target token, and the context vector with a single maxout layer (Goodfellow et al., 2013) of size l and using a softmax over the target language vocabulary (Bahdanau et al., 2015). Fig. 10 illustrates the complete RNNsearch model.

6.4 Convolutional Neural Machine Translation

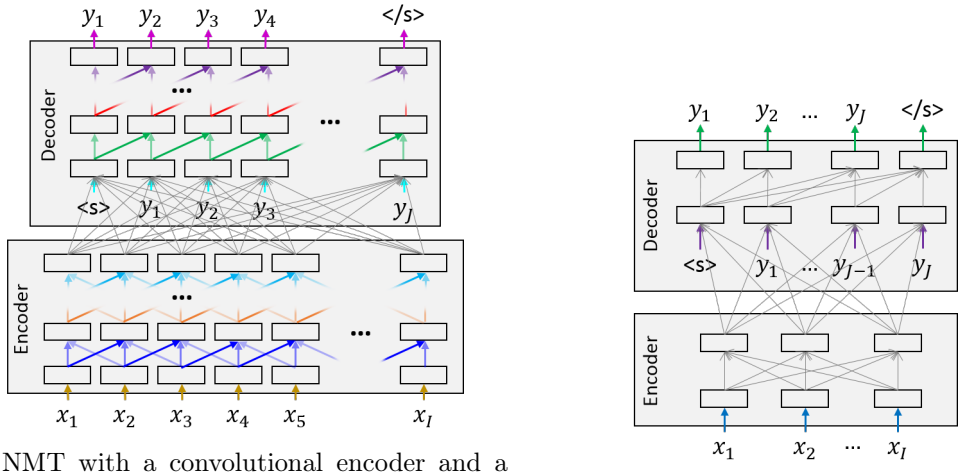
Although convolutional neural networks (CNNs) have first been proposed by Waibel et al. (1989) for phoneme recognition, their traditional use case is computer vision (LeCun et al., 1989, 1990, 1998). CNNs are especially useful for processing images because of two reasons. First, they use a high degree of weight tying and thus reduce the number of parameters dramatically compared to fully connected networks. This is crucial for high dimensional input like visual imagery. Second, they automatically learn space invariant features. Spatial invariance is desirable in vision since we often aim to recognize objects or features regardless of their exact position in the image. In NLP, convolutions are usually one dimensional since we are dealing with sequences rather than two dimensional images as in computer vision. We will therefore limit our discussions to the one dimensional case. We will also exclude concepts like pooling or strides as they are uncommon for sequence models in NLP.

The input to a 1D convolutional layer is a sequence of M-dimensional vectors $u_1, \dots, u_I$. The literature about CNNs usually refers to the M dimensions in each $u_i \in \mathbb{R}^M$ ($i \in [1, I]$) as *channels*, and to the i-axis as *spatial dimension*. The convolution transforms the input sequence $u_1, \dots, u_I$ to an output sequence of N-dimensional $v_1, \dots, v_I$ of the same length by moving a *kernel* of width K over the input sequence. The kernel is a linear transform which maps the K-gram $u_i, \dots, u_{i+K-1}$ to the output v_i for $i \in [1, I]$ (we append $K-1$ padding symbols to the input). Standard convolution parameterizes this linear transform with a full weight matrix $W^{\text{std}} \in \mathbb{R}^{KM \times N}$:

$$\text{StdConv:}(v_i)_n = \sum_{m=1}^{M} \sum_{k=0}^{K-1} W^{\text{std}}_{kM+m,n} (u_{i+k})_m \tag{21}$$

Name	Number of parameters
Standard convolution	KMN
Pointwise convolution	MN
Depthwise convolution	KN
Depthwise separable convolution	N(M+K)

Table 3: Types of convolution and their number of parameters.



(a) NMT with a convolutional encoder and a convolutional decoder like in the ConvS2S architecture (Gehring et al., 2017b).

(b) Purely attention-based NMT as proposed by Vaswani et al. (2017) with two layers.

Figure 12: Convolutional and purely attention-based architectures. The color coding indicates weight sharing. Gray arrows represent attention.

with $i \in [1, I]$ and $n \in [1, N]$. Standard convolution represents two kinds of dependencies: Spatial dependency (inner sum in Eq. 21) and cross-channel dependency (outer sum in Eq. 21). Pointwise and depthwise convolution factor out these dependencies into two separate operations:

$$\text{PointwiseConv:}(v_i)_n = \sum_{m=1}^{M} W^{\text{pw}}_{m,n}(u_i)_m = u_i W^{\text{pw}} \tag{22}$$

$$\text{DepthwiseConv:}(v_i)_n = \sum_{k=0}^{K-1} W^{\text{dw}}_{k,n}(u_{i+k})_n \tag{23}$$

where $W^{\text{pw}} \in \mathbb{R}^{M \times N}$ and $W^{\text{dw}} \in \mathbb{R}^{K \times N}$ are weight matrices. Fig. 11 illustrates the differences between these types of convolution. The idea behind *depthwise separable* convolution is to replace standard convolutional (Eq. 21) with depthwise convolution followed by pointwise convolution. As shown in Tab. 3, the decomposition into two simpler steps reduces the number of parameters and has been shown to make more efficient use of the parameters than regular convolution in vision (Chollet, 2017; Howard et al., 2017).

Using convolution rather than recurrence in NMT models has several potential advantages. First, they reduce sequential computation and are therefore easier to parallelize on

GPU hardware. Second, their hierarchical structure connects distant words via a shorter path than sequential topologies (Gehring et al., 2017b) which eases learning (Hochreiter et al., 2001). Both regular (Kalchbrenner et al., 2016; Gehring et al., 2017b, 2017a) and depthwise separable (Kaiser et al., 2017; Wu et al., 2019) convolution have been used for NMT in the past. Fig. 12a shows the general architecture for a fully convolutional NMT model such as ConvS2S (Gehring et al., 2017b) or SliceNet (Kaiser et al., 2017) in which both encoder and decoder are convolutional. Stacking multiple convolutional layers increases the effective context size which is needed for the translation of long sentences. Therefore, convolutional models are comparably deeper, hence often more difficult to train (Chen et al., 2018a). In the decoder, we need to mask the receptive field of the convolution operations to make sure that the network has no access to future information (van den Oord et al., 2016). Encoder and decoder are connected via attention. Gehring et al. (2017b) used attention into the encoder representations after each convolutional layer in the decoder.

6.5 Self-Attention-Based Neural Machine Translation

Recall that Eq. 5 states that NMT factorizes $P(\mathbf{y}|\mathbf{x})$ into conditionals $P(y_j|y_1^{j-1}, \mathbf{x})$. We have reviewed two ways to model the dependency on the source sentence $\mathbf{x}$ in NMT: via a fixed-length sentence encoding $c(\mathbf{x})$ (Sec. 5) or via time-dependent context vectors $c_j(\mathbf{x})$ which are computed using attention (Sec. 6.1). We have also presented two ways to implement the dependency on the target sentence prefix y_1^{j-1}: via a recurrent connection which passes through the decoder state to the next time step (Sec. 6.3) or via convolution (Sec. 6.4). A third option to model target side dependency is using *self-attention*. Using the terminology introduced in Sec. 6.1, decoder self-attention derives all three components (queries, keys, and values) from the decoder state. The decoder conditions on the translation prefix y_1^{j-1} by attending to its own states from previous time steps. Besides machine translation, self-attention has been applied to various NLP tasks such as sentiment analysis (Cheng et al., 2016a), natural language inference (Shen et al., 2018a; Parikh et al., 2016; Liu et al., 2016; Shen et al., 2018b), text summarization (Paulus et al., 2017), headline generation (Daniil et al., 2019), sentence embedding (Lin et al., 2017; Wu et al., 2018; Zhang et al., 2018a), and reading comprehension (Hu et al., 2018). Similarly to convolution, self-attention introduces short paths between distant words and reduces the amount of sequential computation. An empirical investigation by Tang et al. (2018a) concludes that these short paths are especially useful for learning strong semantic feature extractors, but less so for modelling long-range subject-verb agreement. Furthermore, short paths in attention-based architectures also improve the gradient flow in the backward pass which helps training.[7] Like in convolutional models we also need to mask future decoder states to prevent conditioning on future tokens (cf. Sec. 6.2).

The general layout for self-attention-based NMT models is shown in Fig. 12b. The first example of this new class of NMT models was the Transformer (Vaswani et al., 2017). The Transformer uses attention for three purposes: 1) within the encoder to enable context-sensitive word representations which depend on the whole source sentence, 2) between the encoder and the decoder as in previous models, and 3) within the decoder to condition on

7. As one reviewer of this article pointed out, this is supported by the necessity of reversal of a sequence by Sutskever et al. (2014), compared to Bahdanau et al. (2015).

the current translation history. The Transformer uses multi-head attention (Sec. 6.1) rather than regular attention. Using multi-head attention has been shown to be essential for the Transformer architecture (Tang et al., 2018a; Chen et al., 2018a).

A challenge in self-attention-based models (and to some extent in convolutional models) is that vanilla attention as introduced in Sec. 6.1 by itself has no notion of order. The key-value pairs in the memory are accessed purely based on the correspondence between key and query (*content-based* addressing) and not based on a location of the key in the memory (*location-based*). This is less of a problem in recurrent NMT (Sec. 6.3) as queries, keys, and values are derived from RNN states and already carry a strong sequential signal due to the RNN topology. In the Transformer architecture, however, recurrent connections are removed in favor of attention. Vaswani et al. (2017) tackled this problem using *positional encodings*. Positional encodings are (potentially partial) functions $\text{PE} : \mathbb{N} \nrightarrow \mathbb{R}^D$ where D is the word embedding size, i.e. they are D-dimensional representations of natural numbers. They are added to the (input and output) word embeddings to make them (and consequently the queries, keys, and values) position-sensitive. Vaswani et al. (2017) stacked sine and cosine functions of different frequencies to implement $\text{PE}(\cdot)$:

$$\text{PE}_{\text{sin}}(n)_d = \begin{cases} \sin(10000^{-\frac{d}{D}} n) & : d \text{ is even} \\ \cos(10000^{-\frac{d}{D}} n) & : d \text{ is odd} \end{cases} \tag{24}$$

for $n \in \mathbb{N}$ and $d \in [1, D]$. Alternatively, positional encodings can be learned in an embedding matrix (Gehring et al., 2017b):

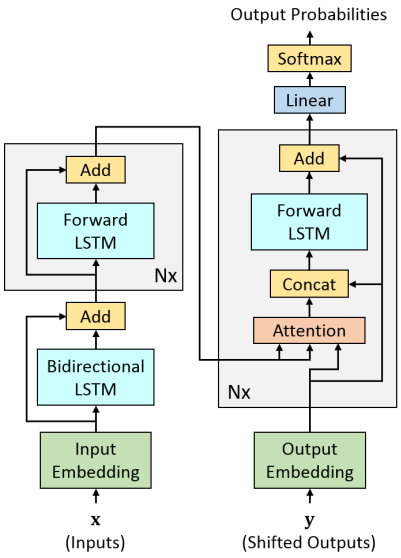
$$\text{PE}_{\text{learned}}(n) = W_{:,n} \tag{25}$$

with weight matrix $W \in \mathbb{R}^{d \times N}$ for some sufficiently large N. The input to $\text{PE}(\cdot)$ is usually the absolute position of the word in the sentence (Vaswani et al., 2017; Gehring et al., 2017b), but relative positioning is also possible (Shaw et al., 2018). A disadvantage of learned positional encodings is that they cannot generalize to sequences longer than N.

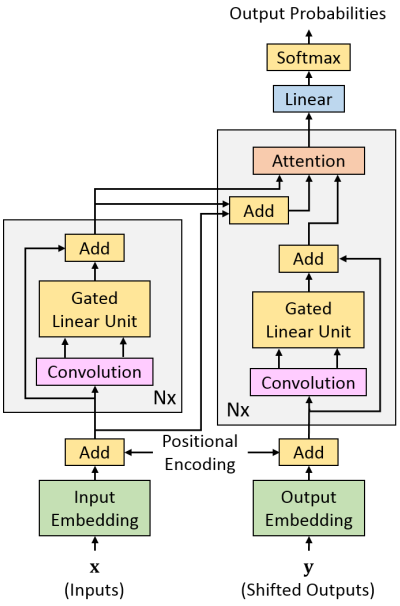
6.6 Comparison of the Fundamental Architectures

As outlined in the previous sections, NMT can come in one of three flavors: recurrent, convolutional, or self-attention-based. In this section, we will discuss three concrete architectures in greater detail – one of each flavor. Fig. 13 visualizes the data streams in Google's Neural Machine Translation system (Wu et al., 2016, GNMT) as an example of a recurrent network, the convolutional ConvS2S model (Gehring et al., 2017b), and the self-attention-based Transformer model (Vaswani et al., 2017) in plate notation. We excluded components like dropout (Srivastava et al., 2014), batch normalization (Ioffe & Szegedy, 2015), and layer normalization (Ba et al., 2016) to simplify the diagrams.

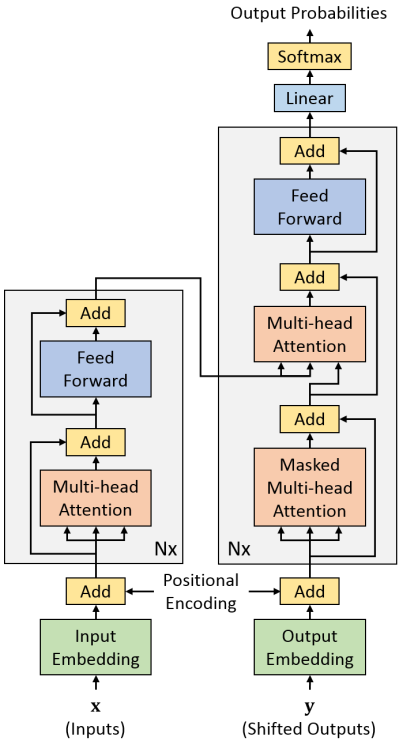
All models fall in the general category of encoder-decoder networks, with the encoder in the left column and the decoder in the right column. Output probabilities are generated by a linear projection layer followed by a softmax activation at the end. They all use attention to connect the encoder with the decoder, although the specifics differ. GNMT (Fig. 13a) uses regular attention, ConvS2S (Fig. 13b) adds the source word encodings to the values, and the Transformer (Fig. 13c) uses multi-head attention (Sec. 6.1). Residual connections (He et al., 2016b) are used in all three architectures to encourage gradient flow in multi-layer networks. Positional encodings are used in ConvS2S and the Transformer, but not in GNMT.



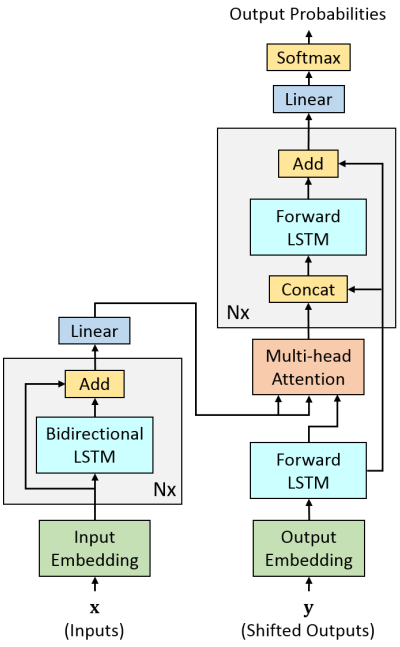
(a) GNMT (Wu et al., 2016).



(b) ConvS2S (Gehring et al., 2017b).



(c) Transformer (Vaswani et al., 2017).



(d) RNMT+ (Chen et al., 2018a).

Figure 13: Comparison of NMT architectures. The three inputs to attention modules are (from left to right): keys (K), values (V), and queries (Q) as in Fig. 7.

An interesting fusion is the RNMT+ model (Chen et al., 2018a) shown in Fig. 13d which reintroduces ideas from the Transformer like multi-head attention into recurrent NMT. Other notable mixed architectures include Gehring et al. (2017a) who used a convolutional encoder with a recurrent decoder, Miculicich et al. (2018), Wang et al. (2019), Werlen et al. (2018) who added self-attention connections to a recurrent decoder, Hao et al. (2019) who used a Transformer encoder and a recurrent encoder in parallel, and Lin et al. (2018) who equipped a recurrent decoder with a convolutional decoder to provide global target-side context. Using recurrence rather than self-attention in the decoder avoids the quadratic inference time complexity as only a single hidden state (not all previous hidden states) has to be passed through to the next timestep. Ablation studies by Tang et al. (2018a), Chen et al. (2018a), Domhan (2018), Tang et al. (2018b), Stahlberg et al. (2018b) provide further insight into the different techniques used across these architectures.

7. Neural Machine Translation Decoding

So far we have described how NMT defines the translation probability $P(\mathbf{y}|\mathbf{x})$. However, in order to apply these definitions directly, both the source sentence $\mathbf{x}$ and the target sentence $\mathbf{y}$ have to be given. They do not directly provide a method for generating a target sentence $\mathbf{y}$ from a given source sentence $\mathbf{x}$ which is the ultimate goal in machine translation. The task of finding the most likely translation $\hat{\mathbf{y}}$ for a given source sentence $\mathbf{x}$ is known as the *decoding* or *inference* problem:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \Sigma_{trg}^*} P(\mathbf{y}|\mathbf{x}). \tag{26}$$

NMT decoding is non-trivial for mainly two reasons. First, the search space is vast as it grows exponentially with the sequence length. For example, if we assume a common vocabulary size of $|\Sigma_{trg}| = 32,000$, there are already more possible translations with 20 words or less than atoms in the observable universe ($32,000^{20} \gg 10^{82}$). Thus, complete enumeration of the search space is impossible. Second, as we will see in Sec. 8, certain types of model errors are very common in NMT. The mismatch between the most likely and the "best" translation has deep implications on search as more exhaustive search often leads to worse translations (Stahlberg & Byrne, 2019). We will discuss possible solutions to both problems in the remainder of Sec. 7.

7.1 Greedy and Beam Search

The most popular decoding algorithms for NMT are greedy search and beam search. Both search procedures are based on the left-to-right factorization of NMT in Eq. 5. Translations are built up from left to right while partial translation prefixes are scored using the conditionals $P(y_j|y_1^{j-1}, \mathbf{x})$. This means that both algorithms work in a time-synchronous manner: in each iteration j, partial hypotheses of (up to) length j are compared to each other, and a subset of them is selected for expansion in the next time step. The algorithms terminate if either all or the best of the selected hypotheses end with the end-of-sentence symbol </s> or if some maximum number of iterations is reached. Fig. 14 illustrates the difference between greedy search and beam search. Greedy search (highlighted in green) selects the single best expansion at each time step: 'c' at $j = 1$, 'a' at $j = 2$, and 'b' at $j = 3$. However, greedy search is vulnerable to the so-called *garden-path problem*: The algorithm selects 'c' in the

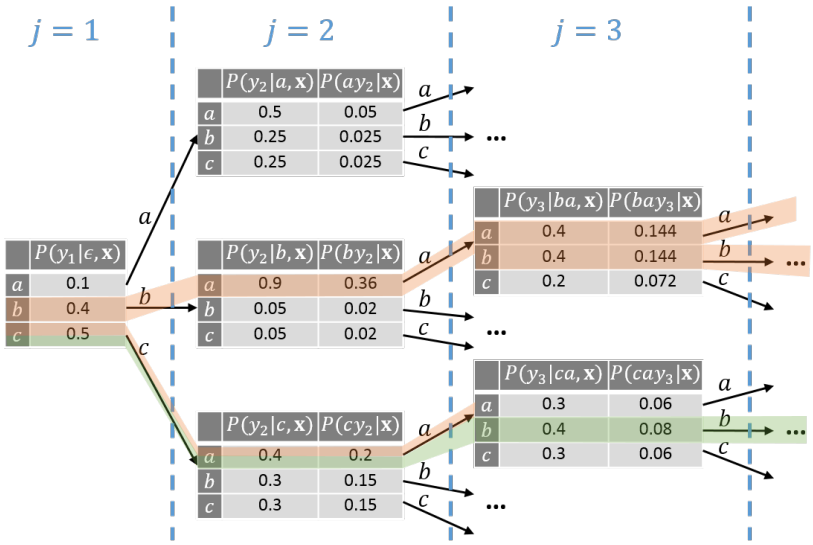


Figure 14: Comparison between greedy (highlighted in green) and beam search (highlighted in orange) with beam size 2.

first time step which turns out to be a mistake later on as subsequent distributions are very smooth and scores are comparably low. However, greedy decoding cannot correct this mistake later as it is already committed to this path. Beam search (highlighted in orange in Fig. 14) tries to mitigate the risk of the garden-path problem by passing not one but n possible translation prefixes to the next time step ($n = 2$ in Fig. 14). The n hypotheses which survive a time step are called *active hypotheses*. At each time step, the accumulated path scores for all possible continuations of active hypotheses are compared, and the n best ones are selected. Thus, beam search not only expands 'c' but also 'b' in time step 1, and thereby finds the high scoring translation prefix 'ba'. Note that although beam search seems to be the more accurate search procedure, it is not guaranteed to always find a translation with higher or equal score as greedy decoding.[8] It is therefore still prone to the garden-path problem, although less so than greedy search. Stahlberg and Byrne (2019) demonstrated that even beam search suffers from a high number of search errors.

7.2 Formal Description of Decoding for the RNNsearch Model

In this section, we will formally define decoding for the RNNsearch model (Bahdanau et al., 2015). We will resort to the mathematical symbols used in Sec. 6.3 to describe the algorithms. First, the source annotations $\mathbf{h}$ are computed and stored as this does not require any search. Then, we compute the distribution for the first target token y_1 using OneStepRNNsearch(s_{init}, <s>, $\mathbf{h}$) (Alg. 1). The initial decoder state s_{init} is often a linear transform of the last encoder hidden state h_I: $s_{init} = Wh_I$ for some weight matrix $W \in \mathbb{R}^{n \times m}$.

8. For example, imagine a series of high entropy conditionals after 'baa' and low entropy conditionals after 'cab' in Fig. 14

Algorithm 1 OneStepRNNsearch($s_{prev}, y_{prev}, \mathbf{h}$)

$\alpha \overset{Eq.\ 16}{\leftarrow} \frac{1}{Z}[\exp(a(s_{prev}, h_i))]_{i \in [1,I]}$ {Attention weights ($\alpha \in \mathbb{R}^I$, Z as in Eq. 16)}
$c \overset{Eq.\ 15}{\leftarrow} \sum_{i=1}^{I} \alpha_i \cdot h_i$ {Context vector update ($c \in \mathbb{R}^m$)}
$s \overset{Eq.\ 17}{\leftarrow} f(s_{prev}, y_{prev}, c)$ {RNN state update ($s \in \mathbb{R}^n$)}
$p \overset{Eq.\ 5}{\leftarrow} g(y_{prev}, s, c)$ {$p \in \mathbb{R}^{|\Sigma_{trg}|}$ is the distribution over the next target token $P(y_j|\cdot)$}
return s, p

Algorithm 2 GreedyRNNsearch($s_{init}, \mathbf{h}$)

$\mathbf{y} \leftarrow \langle\rangle$
$s \leftarrow s_{init}$
$y \leftarrow$ <s>
while $y \neq$ </s> **do**
  $s, p \leftarrow$ OneStepRNNsearch$(s, y, \mathbf{h})$
  $y \leftarrow \arg\max_{w \in \Sigma_{trg}} \pi_w(p)$
  $\mathbf{y}$.append(y)
end while
return $\mathbf{y}$

Algorithm 3 BeamRNNsearch($s_{init}, \mathbf{h}, n \in \mathbb{N}_+$)

$\mathcal{H}_{cur} \leftarrow \{(\epsilon, 0.0, s_{init})\}$ {Initialize with empty translation prefix and zero score}
repeat
  $\mathcal{H}_{next} \leftarrow \emptyset$
  for all $(\mathbf{y}, p_{acc}, s) \in \mathcal{H}_{cur}$ **do**
    if $y_{|\mathbf{y}|} =$ </s> **then**
      $\mathcal{H}_{next} \leftarrow \mathcal{H}_{next} \cup \{(\mathbf{y}, p_{acc}, s)\}$ {Hypotheses ending with </s> are not extended}
    else
      $s, p \leftarrow$ OneStepRNNsearch$(s, y_{|\mathbf{y}|}, \mathbf{h})$
      $\mathcal{H}_{next} \leftarrow \mathcal{H}_{next} \cup \bigcup_{w \in \Sigma_{trg}} (\mathbf{y} \cdot w, p_{acc}\pi_w(p), s)$ {Add all possible continuations}
    end if
  end for
  $\mathcal{H}_{cur} \leftarrow \{(\mathbf{y}, p_{acc}, s) \in \mathcal{H}_{next} : |\{(\mathbf{y}', p'_{acc}, s') \in \mathcal{H}_{next} : p'_{acc} > p_{acc}\}| < n\}$ {Select n-best}
  $(\hat{\mathbf{y}}, \hat{p}_{acc}, \hat{s}) \leftarrow \arg\max_{(\mathbf{y}, p_{acc}, s) \in \mathcal{H}_{cur}} p_{acc}$
until $\hat{y}_{|\hat{\mathbf{y}}|} =$ </s>
return $\hat{\mathbf{y}}$

Greedy decoding selects the most likely target token according to the returned distribution and iteratively calls OneStepRNNsearch$(\cdot)$ until the end-of-sentence symbol </s> is emitted (Alg. 2). We use the projection function $\pi_w(p)$ (Eq. 3) which maps the posterior vector $p \in \mathbb{R}^{|\Sigma_{trg}|}$ to the w-th component.

The beam search strategy (Alg. 3) not only keeps the single best partial hypothesis but a set of n promising hypotheses where n is the size of the beam. A partial hypothesis is represented by a 3-tuple $(\mathbf{y}, p_{acc}, s)$ with the translation prefix $\mathbf{y} \in \Sigma^*_{trg}$, the accumulated score $p_{acc} \in \mathbb{R}$, and the last decoder state $s \in \mathbb{R}^n$.

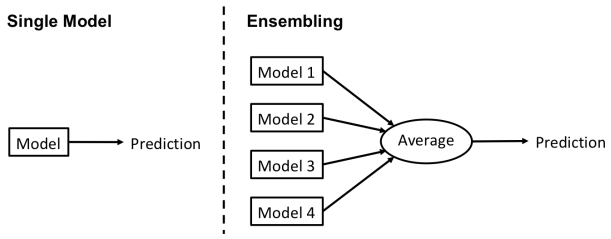


Figure 15: Ensembling four NMT models.

7.3 Ensembling

Ensembling (Dietterich, 2000; Hansen & Salamon, 1990) is a simple yet very effective technique to improve the accuracy of NMT. The basic idea is illustrated in Fig. 15. The decoder makes use of K NMT networks rather than only one which are either trained independently (Sutskever et al., 2014; Neubig, 2016; Wu et al., 2016) or share some amount of training iterations (Sennrich et al., 2016a; Cromieres et al., 2016; Durrani et al., 2016). The ensemble decoder computes predictions for each of the individual models which are then combined using the arithmetic (Sutskever et al., 2014) or geometric (Cromieres et al., 2016) average:

$$S_{\text{arith}}(y_j|y_1^{j-1}, \mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K} P_k(y_j|y_1^{j-1}, \mathbf{x}) \tag{27}$$

$$S_{\text{geo}}(y_j|y_1^{j-1}, \mathbf{x}) = \sum_{k=1}^{K} \log P_k(y_j|y_1^{j-1}, \mathbf{x}). \tag{28}$$

Both $S_{\text{arith}}(\cdot)$ and $S_{\text{geo}}(\cdot)$ can be used as drop-in replacement for the conditionals $P(y_j|y_1^{j-1}, \mathbf{x})$ in Eq. 5. The arithmetic average is more sound as $S_{\text{arith}}(\cdot)$ still forms a valid probability distribution which sums up to one. However, the geometric average $S_{\text{arith}}(\cdot)$ is numerically more stable as log-probabilities can be directly combined without converting them to probabilities. Another difference is that the geometric average favors consensus of the models while the arithmetic average favors the most confident model. Note that the core idea of ensembling is similar to language model interpolation used in statistical machine translation or speech recognition.

Ensembling consistently outperforms single NMT by a large margin. All top systems in recent machine translation evaluation campaigns ensemble a number of NMT systems (Bojar et al., 2016, 2017, 2018, 2019; Sennrich et al., 2016a, 2017a; Neubig, 2016; Cromieres et al., 2016; Durrani et al., 2016; Stahlberg et al., 2018b; Wang et al., 2017; Junczys-Dowmunt, 2018b; Wang et al., 2018b), perhaps most famously taken to the extreme by the WMT18 submission of Tencent that ensembled up to 72 translation models (Wang et al., 2018b). However, the decoding speed is significantly worse since the decoder needs to apply K NMT models rather than only one. This means that the decoder has to perform K more forward passes through the networks, and has to apply the expensive softmax function K more times in each time step. Ensembling also often increases the number of CPU/GPU switches and the communication overhead between CPU and GPU when averaging is implemented on the CPU. Ensembling is also often more difficult to implement than single system NMT.

Knowledge distillation (Buciluǎ et al., 2006; Kim & Rush, 2016; Zhang et al., 2018; Freitag et al., 2017) is one method to deal with the shortcomings of ensembling. Stahlberg and Byrne (2017) proposed to unfold the ensemble into a single network and shrink the unfolded network afterwards for efficient ensembling.

In NMT, all models in an ensemble usually have the same size and topology and are trained on the same data. They differ only due to the random weight initialization and the randomized order of the training samples. Notable exceptions include Freitag and Al-Onaizan (2016) who use ensembling to prevent over-fitting in domain adaptation, He et al. (2018) who combined models that selected their training data based on marginal likelihood, and the UCAM submission to WMT18 (Stahlberg et al., 2018b) that ensembled different NMT architectures with each other.

When all models are equally powerful and are trained with the same data, it is surprising that ensembling is so effective. One common narrative is that different models make different mistakes, but the mistake of one model can be outvoted by the others in the ensemble (Rokach, 2010). This explanation is plausible for NMT since translation quality can vary widely between training runs (Sennrich et al., 2016c). The variance in translation performance may also indicate that the NMT error surface is highly non-convex such that the optimizer often ends up in local optima.[9] Ensembling might mitigate this problem. Ensembling may also have a regularization effect on the final translation scores (Goodfellow et al., 2016).

Checkpoint averaging (Junczys-Dowmunt et al., 2016b, 2016a) is a technique which is often discussed in conjunction with ensembling (Liu et al., 2018b). Checkpoint averaging keeps track of the few most recent checkpoints during training, and averages their weight matrices to create the final model. This results in a single model and thus does not increase the decoding time. Therefore, it has become a very common technique in NMT (Vaswani et al., 2017; Popel & Bojar, 2018; Stahlberg et al., 2018b). Checkpoint averaging addresses a quite different problem than ensembling as it mainly smooths out minor fluctuations in the training curve which are due to the optimizer's update rule or noise in the gradient estimation due to mini-batch training. In contrast, the weights of independently trained models are very different from each other, and there is no obvious direct correspondence between neuron activities across the models. Therefore, checkpoint averaging cannot be applied to independently trained models.

7.4 Decoding Direction

Standard NMT factorizes the probability $P(\mathbf{y}|\mathbf{x})$ from left to right (L2R) according Eq. 5. Mathematically, the left-to-right order is rather arbitrary, and other arrangements such as a right-to-left (R2L) factorization are equally correct:

$$P(\mathbf{y}|\mathbf{x}) = \underbrace{\prod_{j=1}^{J} P(y_j|y_1^{j-1},\mathbf{x})}_{=P(y_1|\mathbf{x})\cdot P(y_2|y_1,\mathbf{x})\cdot P(y_3|y_1,y_2,\mathbf{x})\cdots} = \underbrace{\prod_{j=1}^{J} P(y_j|y_{j+1}^{J},\mathbf{x})}_{=P(y_J|\mathbf{x})\cdot P(y_{J-1}|y_J,\mathbf{x})\cdot P(y_{J-2}|y_{J-1},y_J,\mathbf{x})\cdots} . \quad (29)$$

9. Another plausible explanation for this variance are search errors as discussed in Sec. 8.

NMT models which produce the target sentence in reverse order have led to some gains in evaluation systems when combined with left-to-right models (Sennrich et al., 2016a; Wang et al., 2017; Stahlberg et al., 2018b; Wang et al., 2018b). A common combination scheme is based on rescoring: A strong L2R ensemble first creates an n-best list which is then rescored with a R2L model (Liu et al., 2016; Sennrich et al., 2016a). Stahlberg et al. (2018b) used R2L models via a minimum Bayes risk framework. The L2R and R2L systems are normally trained independently, although some recent work proposes joint training schemes in which each direction is used as a regularizer for the other direction (Zhang et al., 2018d; Yang et al., 2018c). Other orderings besides L2R and R2L have also been proposed such as middle-out (Mehri & Sigal, 2018), top-down in a binary tree (Welleck et al., 2019), insertion-based (Gu et al., 2019; Stern et al., 2019; Östling & Tiedemann, 2017; Gu et al., 2019), or in source sentence order (Stahlberg et al., 2018).

7.5 Efficiency

NMT decoding is very fast on GPU hardware and can reach up to 5000 words per second.[10] However, GPUs are very expensive, and speeding up CPU decoding to the level of SMT remains more challenging. Therefore, how to improve the efficiency of neural sequence decoding algorithms is still an active research question. One bottleneck is the sequential left-to-right order of beam search which makes parallelization difficult. Stern et al. (2018) suggested to compute multiple time steps in parallel and validate translation prefixes afterwards. Kaiser et al. (2018) reduced the amount of sequential computation by learning a sequence of latent discrete variables which is shorter than the actual target sentence, and generating the final sentence from this latent representation in parallel. Di Gangi and Federico (2018) sped up recurrent NMT by using a simplified architecture for recurrent units. Another line of research tries to reintroduce the idea of *hypothesis recombination* to neural models. This technique is used extensively in traditional SMT (see Koehn, 2010 for an overview). The idea is to keep only the better of two partial hypotheses if it is guaranteed that both will be scored equally in the future. For example, this is the case for n-gram language models if both hypotheses end with the same n-gram. The problem in neural sequence models is that they condition on the full translation history. Therefore, hypothesis recombination for neural sequence models does not insist on exact equivalence but clusters hypotheses based on the similarity between RNN states or the n-gram history (Zhang et al., 2018e; Liu et al., 2014). A similar idea was used by Lecorvé and Motlicek (2012) to approximate RNNs with WFSTs which also requires mapping histories into equivalence classes.

It is also possible to speed up beam search by reducing the beam size. Wu et al. (2016), Freitag and Al-Onaizan (2017) suggested to use a variable beam size, using various heuristics to decide the beam size at each time step. Alternatively, the NMT model training can be tailored towards the decoding algorithm (Goyal et al., 2018; Wiseman & Rush, 2016; Collobert et al., 2019; Gu et al., 2017b). Wiseman and Rush (2016) proposed a loss function for NMT training which penalizes when the reference falls off the beam during training. Kim and Rush (2016) reported that knowledge distillation (Buciluǎ et al., 2006) reduces the gap between greedy decoding and beam decoding significantly. Greedy decoding can also be

10. `https://marian-nmt.github.io/features/`

improved by using a small actor network which modifies the hidden states in an already trained model (Gu et al., 2017b; Chen et al., 2018b).

Non- or partially autoregressive NMT which aims to reduce or remove the sequential dependency on the translation prefix inside the decoder for enhanced parallelizability has been studied by Wang et al. (2018a), Gu et al. (2017a), Guo et al. (2018), Wang et al. (2019), Libovický and Helcl (2018), Lee et al. (2018), Akoury et al. (2019).

7.6 Generating Diverse Translations

An issue with using beam search is that the hypotheses found by the decoder are very similar to each other and often differ only by one or two words (Li & Jurafsky, 2016; Li et al., 2016b; Gimpel et al., 2013). The lack of diversity is problematic for several reasons. First, natural language in general and translation in particular often come with a high level of ambiguity that is not represented well by non-diverse n-best lists. Second, it impedes user interaction as NMT is not able to provide the user with alternative translations if needed. Third, collecting statistics about the search space such as estimating the probabilities of n-grams for minimum Bayes-risk decoding (Goel et al., 2000; Kumar & Byrne, 2004; Tromble et al., 2008; Iglesias et al., 2018; Stahlberg et al., 2018b, 2017) or risk-based training (Shen et al., 2016) is much less effective.

Cho (2016) added noise to the activations in the hidden layer of the decoder network to produce alternative high scoring hypotheses. This is justified by the observation that small variations of a hidden configuration encode semantically similar context (Bengio et al., 2013). Li and Jurafsky (2016), Li et al. (2016b) proposed a diversity promoting modification of the beam search objective function. They added an explicit penalization term to the NMT score based on a maximum mutual information criterion which penalizes hypotheses from the same parent node. Note that both extensions can be used together (Cho, 2016). Vijayakumar et al. (2016) suggested to partition the active hypotheses in groups, and use a dissimilarity term to ensure diversity between groups. Park et al. (2016) found alternative translations by k-nearest neighbor search from the greedy translation in a translation memory. However, none of these techniques have been adopted widely in production systems.

8. NMT Model Errors

NMT is highly effective in assigning scores (or probabilities) to translations because, in stark contrast to SMT, it does not make any conditional independence assumptions in Eq. 5 to model sentence-level translation. A potential drawback of such a powerful model is that it prohibits the use of sophisticated search procedures. Compared to hierarchical SMT systems like Hiero (Chiang, 2007) that explore very large search spaces, NMT beam search appears to be overly simplistic. This observation suggests that translation errors in NMT are more likely due to *search errors* (the decoder does not find the highest scoring translation) than *model errors* (the model assigns a higher probability to a worse translation). Interestingly, this is not necessarily the case. Search errors in NMT have been studied by Niehues et al. (2017), Stahlberg et al. (2018), Stahlberg and Byrne (2019). In particular, Stahlberg and Byrne (2019) demonstrated the high number of search errors in NMT decoding. However, as we will show in this section, NMT also suffers from various kinds of model errors in practice despite its theoretical advantage.

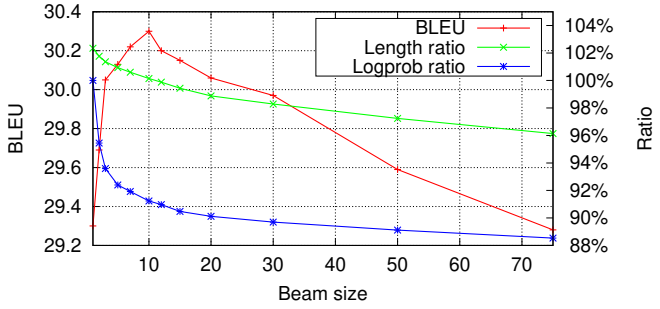


Figure 16: Performance of a Transformer model on English-German (WMT15) under varying beam sizes. The BLEU score peaks at beam size 10, but then suffers from a length ratio (hypothesis length / reference length) below 1. The log-probabilities are shown as a ratio with respect to greedy decoding.

8.1 Sentence Length

Increasing the beam size exposes one of the most noticeable model errors in NMT. The red curve in Fig. 16 plots the BLEU score (Papineni et al., 2002) of a recent Transformer NMT model against the beam size. A beam size of 10 is optimal on this test set. Wider beams lead to a steady drop in translation performance because the generated translations are becoming too short (green curve). However, as expected, the log-probabilities of the found translations (blue curve) are decreasing as we increase the beam size. NMT seems to assign too much probability mass to short hypotheses which are only found with more exhaustive search. Sountsov and Sarawagi (2016) argue that this model error is due to the locally normalized maximum likelihood training objective in NMT that underestimates the margin between the correct translation and shorter ones if trained with regularization and finite data. A similar argument was made by Murray and Chiang (2018) who pointed out the difficulty for a locally normalized model to estimate the "budget" for all remaining (longer) translations in each time step. Kumar and Sarawagi (2019) demonstrated that NMT models are often poorly calibrated, and that calibration issues can cause the length deficiency in NMT. A similar case is illustrated in Fig. 17. The NMT model underestimates the combined probability mass of translations continuing after "Stadtrat" in time step 7 and overestimates the probability of the period symbol. Greedy decoding does not follow the green translation since "der" is more likely in time step 7. However, beam search with a large beam keeps the green path and thus finds the shorter (incomplete) translation with better score. In fact, Stahlberg and Byrne (2019) linked the bias of large beam sizes towards short translations with the reduction of search errors.

At first glance this seems to be good news: fast beam search with a small beam size is already able to find good translations. However, fixing the model error of short translations by introducing search errors with a narrow beam seems like fighting fire with fire. In practice, this means that the beam size is yet another hyper-parameter which needs to be tuned for each new NMT training technique (eg. label smoothing (Szegedy et al., 2016) usually requires a larger beam), NMT architecture (the Transformer model is usually decoded with a smaller beam than typical recurrent models), and language pair (Koehn & Knowles, 2017). More

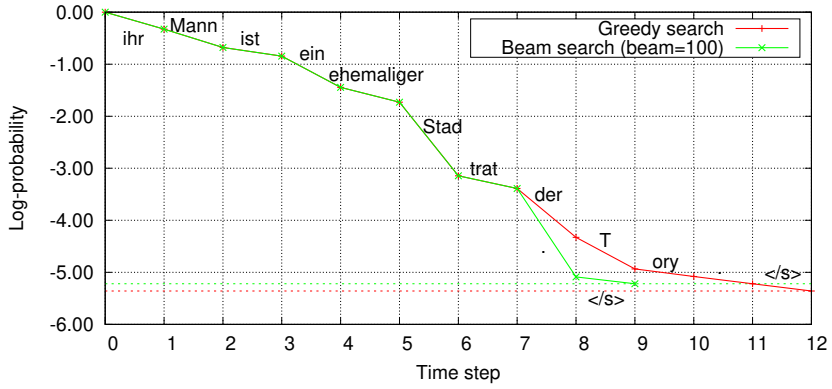


Figure 17: The length deficiency in NMT translating the English source sentence "Her husband is a former Tory councillor." into German following Murray and Chiang (2018). The NMT model assigns a better score to the short translation "Ihr Mann ist ein ehemaliger Stadtrat." than to the greedy translation "Ihr Mann ist ein ehemaliger Stadtrat der Tory." even though it misses the former affiliation of the husband with the Tory Party.

importantly, it is not clear whether there are gains to be had from reducing the number of search errors with wider beams which are simply obliterated by the NMT length deficiency.

8.1.1 Model-Agnostic Length Models

The first class of approaches to alleviate the length problem is model-agnostic. Methods in this class treat the NMT model as a black box but add a correction term to the NMT score to bias beam search towards longer translations. A simple method is called *length normalization* which divides the NMT probability by the sentence length (Jean et al., 2015b; Boulanger-Lewandowski et al., 2013):

$$S_{\text{LN}}(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|} \tag{30}$$

Wu et al. (2016) proposed an extension of this idea by introducing a tunable parameter α:

$$S_{\text{LN-GNMT}}(\mathbf{y}|\mathbf{x}) = \log P(\mathbf{y}|\mathbf{x}) \frac{(1+5)^\alpha}{(1+|\mathbf{y}|)^\alpha} \tag{31}$$

Alternatively, like in SMT we can use a word penalty $\gamma(j, \mathbf{x})$ which rewards each word in the sentence:

$$S_{\text{WP}}(\mathbf{y}|\mathbf{x}) = \sum_{j=1}^{J} \gamma(j, \mathbf{x}) + \log P(y_j | y_1^{j-1}, \mathbf{x}) \tag{32}$$

A constant reward which is independent of $\mathbf{x}$ and j can be found with minimum-error-rate-training (He et al., 2016c) or with a gradient-based learning scheme (Murray & Chiang, 2018). Alternative policies which reward words with respect to some estimated sentence length were suggested by Huang et al. (2017), Yang et al. (2018b).

8.1.2 Source-Side Coverage Models

Tu et al. (2016) connected the sentence length issue in NMT with the lack of an explicit mechanism to check the source-side coverage of a translation. Traditional SMT keeps track of a coverage vector $\mathcal{C}_{\text{SMT}} \in \{0,1\}^I$ which contains 1 for source words which are already translated and 0 otherwise. $\mathcal{C}_{\text{SMT}}$ is used to guard against *under-translation* (missing translations of some words) and *over-translation* (some words are unnecessarily translated multiple times). Since vanilla NMT does not use an explicit coverage vector it can be prone to both under- and over-translation (Tu et al., 2016; Yang et al., 2018a) and tends to prefer fluency over adequacy (Kong et al., 2018). There are two popular ways to model coverage in NMT, both make use of the encoder-decoder attention weight matrix A introduced in Sec. 6.1. The simpler methods combine the scores of an already trained NMT system with a coverage penalty $cp(\mathbf{x}, \mathbf{y})$ without retraining. This penalty represents how much of the source sentence is already translated. Wu et al. (2016) proposed the following term:

$$cp(\mathbf{x}, \mathbf{y}) = \beta \sum_{i=1}^{I} \log\big(\min(\sum_{j=1}^{J} A_{i,j}, 1.0)\big). \tag{33}$$

A very similar penalty was suggested by Li et al. (2018):

$$cp(\mathbf{x}, \mathbf{y}) = \alpha \sum_{i=1}^{I} \log\big(\max(\sum_{j=1}^{J} A_{i,j}, \beta)\big) \tag{34}$$

where α and β are hyper-parameters that are tuned on the development set.

An even tighter integration can be achieved by changing the NMT architecture itself and jointly training it with a coverage model (Tu et al., 2016; Mi et al., 2016a). Tu et al. (2016) reintroduced an explicit coverage matrix $\mathcal{C} \in [0,1]^{I \times J}$ to NMT. Intuitively, the j-th column $\mathcal{C}_{:,j}$ stores to what extent each source word has been translated in time step j. $\mathcal{C}$ can be filled with an RNN-based controller network (the "neural network based" coverage model of Tu et al. (2016)). Alternatively, we can directly use A to compute the coverage (the "linguistic" coverage model of Tu et al. (2016)):

$$\mathcal{C}_{i,j} = \frac{1}{\Phi_i} \sum_{k=1}^{j} A_{i,k} \tag{35}$$

where Φ_i is the estimated number of target words the i-th source word generates which is similar to fertility in SMT. Φ_i is predicted by a feedforward network that conditions on the i-th encoder state. In both the neural network based and the linguistic coverage model, the decoder is modified to additionally condition on $\mathcal{C}$. The idea of using fertilities to prevent over- and under-translation has also been explored by Malaviya et al. (2018). A coverage model for character-based NMT was suggested by Kazimi and Costa-Jussá (2017).

All approaches discussed in this section operate on the attention weight matrix A and are thus only readily applicable to models with single encoder-decoder attention like GNMT, but not to models with multiple encoder-decoder attention modules such as ConvS2S or the Transformer (see Sec. 6.6 for detailed descriptions of GNMT, ConvS2S, and the Transformer).

Vocabulary size	Number of parameters		
	Embeddings	Rest	Total
30K	55.8M	27.9M	83.7M
50K	93.1M	27.9M	121.0M
150K	279.2M	27.9M	307.1M

Table 4: Number of parameters in the original RNNsearch model (Bahdanau et al., 2015) as presented in Sec. 6.3 (1000 hidden units, 620-dimensional embeddings). The model size highly depends on the vocabulary size.

8.1.3 Controlling Mechanisms for Output Length

In some sequence prediction tasks such as headline generation or text summarization, the approximate desired output length is known in advance. In such cases, it is possible to control the length of the output sequence by explicitly feeding in the desired length to the neural model. The length information can be provided as additional input to the decoder network (Fan et al., 2018; Liu et al., 2018a), at each time step as the number of remaining tokens (Kikuchi et al., 2016), or by modifying Transformer positional embeddings (Takase & Okazaki, 2019). However, these approaches are not directly applicable to machine translation as the translation length is difficult to predict with sufficient accuracy.

9. Open Vocabulary Neural Machine Translation

As discussed in Sec. 3, NMT and other neural NLP models use embedding matrices to represent words as real-valued vectors. Embedding matrices need to have a fixed shape to make joint training with the translation model possible, and thus can only be used with a fixed and pre-defined vocabulary. This has several major implications for NMT.

9.1 Using Large Output Vocabularies

One problem with large output vocabularies is that the size of the embedding matrices grows with the vocabulary size. As shown in Tab. 4, the embedding matrices make up most of the model parameters of a standard RNNsearch model. Increasing the vocabulary size inflates the model drastically. Large models require a small batch size because they take more space in the (GPU) memory, but reducing the batch size often leads to noisier gradients, slower training, and eventually worse model performance (Popel & Bojar, 2018). Furthermore, a large softmax output layer is computationally very expensive. In contrast, traditional (symbolic) MT systems can easily use very large vocabularies (Heafield et al., 2013; Lin & Dyer, 2010; Chiang, 2007; Koehn, 2010). Besides these practical issues, training embedding matrices for large vocabularies is also complicated by the long-tail distribution of words in a language. Zipf's law (Zipf, 1946) states that the frequency of any word and its rank in the frequency table are inversely proportional to each other. Fig. 18 shows that 843K of the 875K distinct words (96.5%) occur less than 100 times in an English text with 140M running words – that is less than 0.00007% of the entire text. It is difficult to train robust word embeddings for such rare words. Word-based NMT models address this issue by restricting the vocabulary to the n most frequent words, and replacing all other words by

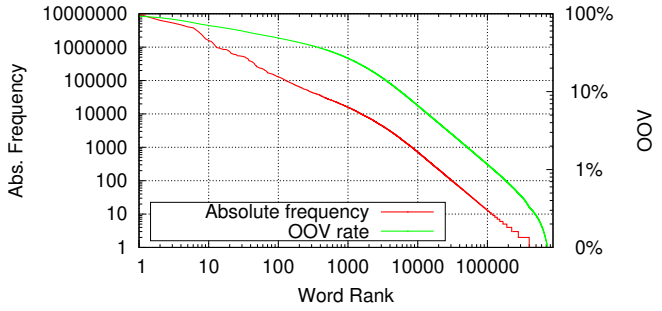


Figure 18: Distribution of words in the English portion of the English-German WMT18 training set (5.9M sentences, 140M words).

a special token UNK. A problem with that approach is that the UNK token may appear in the generated translation. In fact, limiting the vocabulary to the 30K most frequent words results in an out-of-vocabulary rate (OOV) of 2.9% on the training set (Fig. 18). That means an UNK token can be expected to occur every 35 words. In practice, the number of UNKs is usually even higher. One simple reason is that the test set OOV rate is often higher than on the training set because the distribution of words and phrases naturally varies across genre, corpora, and time. Another observation is that word-based NMT often prefers emitting UNK even if a more appropriate word is in the NMT vocabulary. This is possibly due to the misbalance between the UNK token and other words: replacing all rare words with the same UNK token leads to an over-representation of UNK in the training set, and therefore a strong bias towards UNK during decoding.

9.1.1 Translation-Specific Approaches

Jean et al. (2015a) distinguished between *translation-specific* and *model-specific* approaches. Translation-specific approaches keep the shortlist vocabulary in the original form, but correct UNK tokens afterwards. For example, the UNK replace technique (Luong et al., 2015b; Le et al., 2016) keeps track of the positions of source sentence words which correspond to the UNK tokens. In a post-processing step, they replaced the UNK tokens with the most likely translation of the aligned source word according to a bilingual word-level dictionary which was extracted from a word-aligned training corpus. Gulcehre et al. (2016) followed a similar idea but used a special pointer network for referring to source sentence words. These approaches are rather ad-hoc because simple dictionary lookup without context is not a very strong model of translation. Li et al. (2016) replaced each OOV word with a similar in-vocabulary word based on the cosine similarity between their distributed representations in a pre-processing step. However, this technique cannot tackle all OOVs as it is based on vector representations of words which are normally only available for a closed vocabulary. Moreover, the replacements might differ from the original meaning significantly. Further UNK replacement strategies were presented by Li et al. (2017, 2017), Miao et al. (2017), but all share the inevitable limitation of all translation-specific approaches, namely that the translation model itself is indiscriminative between a large number of OOVs.

9.1.2 Model-Specific Approaches

Model-specific approaches change the NMT model to make training with large vocabularies feasible. For example, Nguyen and Chiang (2018) improved the translation of rare words in NMT by adding a lexical translation model which directly connects corresponding source and target words. Another very popular idea is to train networks to output probability distributions without using the full softmax (Andreas & Klein, 2015). Noise-contrastive estimation (Gutmann & Hyvärinen, 2010; Dyer, 2014, NCE) trains a logistic regression model which discriminates between real training examples and noise. For example, to train an embedding for a word w, Mnih and Kavukcuoglu (2013) treat w as a positive example, and sample from the global unigram word distribution in the training data to generate negative examples. The logistic regression model is a binary classifier and thus does not need to sum over the full vocabulary. NCE has been used to train large vocabulary neural sequence models such as language models (Mnih & Teh, 2012). The technique falls into the category of self-normalizing training (Andreas & Klein, 2015) because the model is trained to emit normalized distributions without explicitly summing over the output vocabulary. Devlin et al. (2014) encouraged the network to learn parameters which generate normalized output by adding the value of the partition function to the training loss.

Another approach (sometimes referred to as *vocabulary selection*) is to approximate the partition function of the full softmax by using only a subset of the vocabulary. This subset can be selected in different ways. For example, Jean et al. (2015a) applied importance sampling to select a small set of words for approximating the partition function. Both softmax sampling and UNK replace have been used in one of the winning systems at the WMT'15 evaluation on English-German (Jean et al., 2015b). Various methods have been proposed to select the vocabulary to normalize over during decoding, such as fetching all possible translations in a conventional phrase table (Mi et al., 2016c), using the vocabulary of the translation lattices from a traditional MT system (Stahlberg et al., 2016), and attention-based (Sankaran et al., 2017) and embedding-based (L'Hostis et al., 2016) methods.

9.2 Character-Based NMT

Arguably, both translation-specific and model-specific approaches to word-based NMT are fundamentally flawed. Translation-specific techniques like UNK replace are indiscriminative between translations that differ only by OOV words. A translation model which assigns exactly the same score to a large number of hypotheses is of limited use by its own. Model-specific approaches suffer from the difficulty of training embeddings for rare words (Sec. 9.1). Compound or morpheme splitting can mitigate this issue to a certain extent (Hans & Milton, 2016; Tamchyna et al., 2017). More importantly, however, a fully-trained NMT system even with a very large vocabulary cannot be extended with new words. Customizing systems to new domains (and thus new vocabularies) is a crucial requirement for commercial MT. Moreover, many OOV words are proper names which can be passed through untranslated. Hiero (Chiang, 2007) and other symbolic systems can easily be extended with new words and phrases.

More recent attempts try to alleviate the vocabulary issue in NMT by departing from words as modelling units. These approaches decompose the word sequences into finer-grained units and model the translation between those instead of words. To the best of

our knowledge, Ling et al. (2015) were the first who proposed an NMT architecture which translates between sequences of characters. The core of their NMT network is still on the word-level, but the input and output embedding layers are replaced with subnetworks that compute word representations from the characters of the word. Such a subnetwork can be recurrent (Ling et al., 2015; Johansen et al., 2016) or convolutional (Costa-jussà & Fonollosa, 2016; Kim et al., 2016). This idea was extended to a hybrid model by Luong and Manning (2016) who used the standard lookup table embeddings for in-vocabulary words and the LSTM-based embeddings only for OOVs.

Having a word-level model at the core of a character-based system does circumvent the closed vocabulary restriction of purely word-based models, but it is still segmentation-dependent: The input text has to be preprocessed with a tokenizer that separates words by blank symbols in languages without word boundary markers, optionally applies compound or morpheme splitting in morphologically rich languages, and isolates punctuation symbols. Since tokenization is by itself error-prone and can degrade the translation performance (Domingo et al., 2018), it is desirable to design character-level systems that do not require any prior segmentation. Chung et al. (2016) used a bi-scale recurrent neural network that is similar to dynamically segmenting the input using jointly learned gates between a slow and a fast recurrent layer. Lee et al. (2017), Yang et al. (2016) used convolution to achieve segmentation-free character-level NMT. Costa-jussà et al. (2017) took character-level NMT one step further and used bytes rather than characters to help multilingual systems. Gulcehre et al. (2017) added a planning mechanism to improve the attention weights between character-based encoders and decoders.

9.3 Subword-Unit-Based NMT

As a compromise between characters and full words, compression methods like Huffman codes (Chitnis & DeNero, 2015), word piece models (Schuster & Nakajima, 2012; Wu et al., 2016), or byte pair encoding (Sennrich et al., 2016c) can be used to transform the words to sequences of subword units. Subwords have been used rarely for traditional SMT (Kunchukuttan & Bhattacharyya, 2017, 2016; Liu et al., 2018), but are currently the most common translation units for NMT. Byte pair encoding (Gage, 1994, BPE) initializes the set of available subword units with the character set of the language. This set is extended iteratively in subsequent merge operations. Each merge combines the two units with the highest number of co-occurrences in the text.[11] This process terminates when the desired vocabulary size is reached. This vocabulary size is often set empirically, but can also be tuned on data (Salesky et al., 2018).

Given a fixed BPE vocabulary, there are often multiple ways to segment an unseen text.[12] The ambiguity stems from the fact that symbols are still part of the vocabulary even after they are merged. Most BPE implementations select a segmentation greedily by preferring longer subword units. Interestingly, the ambiguity can also be used as source of noise for regularization. Kudo (2018) reported large gains by augmenting the training data

11. Wu and Zhao (2018) proposed alternatives to the co-occurrence counts. The wordpiece model (Schuster & Nakajima, 2012; Wu et al., 2016) can also be seen as replacing the co-occurrence counts with a language model objective.
12. This is not true for other subword compression algorithms. For example, Huffman codes (Chitnis & DeNero, 2015) are prefix codes and thus unique.

Character-based NMT	Subword-based NMT
+ Better at transliteration (Sennrich, 2017). + Dynamic segmentation favors characters (Kreutzer & Sokolov, 2018). + More robust against noise (Durrani et al., 2018; Belinkov & Bisk, 2017). + Better modelling of morphology (Durrani et al., 2018). + Character-level decoders better than subword-based ones in some studies (Chung et al., 2016; Cherry et al., 2018). − Character-based NMT computationally more expensive than subword-based NMT (Cherry et al., 2018). − More prone to vanishing gradients (Chung et al., 2016). − Long-range dependencies have to be modelled over longer time-spans (Lee et al., 2017).	+ More grammatical (Sennrich, 2017). + Iterative BPE segmentation favors larger vocabulary sizes (Salesky et al., 2018). + Better at syntax (Durrani et al., 2018). + Tends to outperform character-based models in recent MT evaluations (Bojar et al., 2016, 2017, 2018).

Table 5: Summary of studies comparing characters and subword-units for neural machine translation.

with alternative subword segmentations and by decoding from multiple segmentations of the same source sentence.

Segmentation approaches differ in the level of constraints they impose on the subwords. A common constraint is that subwords cannot span over multiple words (Sennrich et al., 2016c). However, enforcing this constraint again requires a tokenizer which is a potential source of errors (see Sec. 9.2). The SentencePiece model (Kudo & Richardson, 2018) is a tokenization-free subword model that is estimated on raw text. On the other side of the spectrum, it has been observed that automatically learned subwords generally do not correspond to linguistic entities such as morphemes, suffixes, affixes etc. However, linguistically-motivated subword units as proposed by Huck et al. (2017), Macháček et al. (2018), Ataman et al. (2017), Pinnis et al. (2017) that also take morpheme boundaries into account do not always improve over completely data-driven ones.

9.4 Words, Subwords, or Characters?

There is no conclusive agreement in the literature whether characters or subwords are the better translation units for NMT. Tab. 5 summarizes some of the arguments. The tendency seems to be that character-based systems have the potential of outperforming subword-based NMT, but they are technically difficult to deploy. Therefore, most systems in the

WMT18 evaluation are based on subwords (Bojar et al., 2018). On a more profound level, we do see the shift towards small modelling units not without some concern. Chung et al. (2016) noted that "we often have a priori belief that a word, or its segmented-out lexeme, is a basic unit of meaning, making it natural to approach translation as mapping from a sequence of source-language words to a sequence of target-language words." Translation is the task of transferring *meaning* from one language to another, and it makes intuitive sense to model this process with meaningful units. The decades of research in traditional SMT were characterized by a constant movement towards larger translation units – starting from the word-based IBM models (Brown et al., 1993) to phrase-based MT (Koehn, 2004) and hierarchical SMT (Chiang, 2007) that models syntactic structures. Expressions consisting of multiple words are even more appropriate units than words for translation since there is rarely a 1:1 correspondence between source and target words. In contrast, the starting point for character- and subword-based models is the language's writing system. Most writing systems are not logographic but alphabetic or syllabaric and thus use symbols without any relation to meaning. The introduction of symbolic word-level and phrase-level information to NMT is one of the main motivations for NMT-SMT hybrid systems (Sec. 13).

10. Using Monolingual Training Data

In practice, parallel training data for MT is hard to acquire and expensive, whereas untranslated monolingual data is usually abundant. This is one of the reasons why language models (LMs) are central to traditional SMT. For example, in Hiero (Chiang, 2007), the translation grammar spans a vast space of possible translations but is weak in assigning scores to them. The LM is mainly responsible for selecting a coherent and fluent translation from that space. However, the vanilla NMT formalism does not allow the integration of an LM or monolingual data in general.[13]

There are several lines of research which investigate the use of monolingual training data in NMT. Gulcehre et al. (2015, 2017) suggested to integrate a separately trained RNN-LM into the NMT decoder. Similarly to traditional SMT (Koehn, 2004) they started out with combining RNN-LM and NMT scores via a log-linear model ('shallow fusion'). They reported even better performance with 'deep fusion' which uses a controller network that dynamically adjusts the weights between RNN-LM and NMT. Both deep fusion and n-best reranking with count-based language models have led to some gains in WMT evaluation systems (Jean et al., 2015b; Wang et al., 2017). The 'simple fusion' technique (Stahlberg et al., 2018a) trains the translation model to predict the residual probability of the training data added to the prediction of a pre-trained and fixed LM.

The second line of research makes use of monolingual text via data augmentation. The idea is to add monolingual data in the target language to the natural parallel training corpus. Different strategies for filling in the source side for these sentences have been proposed such as using a single dummy token (Sennrich et al., 2016b) or copying the target sentence over to the source side (Currey et al., 2017). The most successful strategy is called back-translation (Schwenk, 2008; Sennrich et al., 2016b) which employs a separate translation system in the reverse direction to generate the source sentences for the monolingual target

13. An exception is the neural noisy channel model of Yu et al. (2016) that uses a language model as the unconditional source model.

language sentences. The back-translating system is usually smaller and computationally cheaper than the final system for practical reasons, although with enough computational resources improving the quality of the reverse system can affect the final translation performance significantly (Burlot & Yvon, 2018). Iterative approaches that back-translate with systems that were by themselves trained with back-translation can yield improvements (Hoang et al., 2018; Niu et al., 2018; Zhang et al., 2018c) although they are not widely used due to their computational costs. Back-translation has become a very common technique and has been used in nearly all neural submissions to recent evaluation campaigns (Sennrich et al., 2016a; Bojar et al., 2017, 2018).

A major limitation of back-translation is that the amount of synthetic data has to be balanced with the amount of real parallel data (Sennrich et al., 2016b, 2016a; Poncelas et al., 2018). Therefore, the back-translation technique can only make use of a small fraction of the available monolingual data. An imbalance between synthetic and real data can be partially corrected by over-sampling – duplicating real training samples a number of times to match the synthetic data size. However, very high over-sampling rates often do not work well in practice. Recently, Edunov et al. (2018a) proposed to add noise to the back-translated sentences to provide a stronger training signal from the synthetic sentence pairs. They showed that adding noise not only improves the translation quality but also makes the training more robust against a high ratio of synthetic against real sentences. The effectiveness of using noise for data augmentation in NMT has also been confirmed by Wang et al. (2018b). These methods increase the variety of the training data and thus make it harder for the model to fit which ultimately leads to stronger training signals. The variety of synthetic sentences in back-translation can also be increased by sampling multiple sentences from the reverse translation model (Imamura et al., 2018).

A third class of approaches changes the NMT training loss function to incorporate monolingual data. For example, Cheng et al. (2016b), Tu et al. (2017), Escolano et al. (2018) proposed to add autoencoder terms to the training objective which capture how well a sentence can be reconstructed from its translated representation. Using the reconstruction error is also central to (unsupervised) dual learning approaches (He et al., 2016a; Hassan et al., 2018; Wang et al., 2018c). However, training with respect to the new loss is often computationally intensive and requires approximations. Alternatively, multi-task learning has been used to incorporate source-side (Zhang & Zong, 2016b) and target-side (Domhan & Hieber, 2017) monolingual data. Another way of utilizing monolingual data in both source and target language is to warm start Seq2Seq training from pre-trained encoder and decoder networks (Ramachandran et al., 2017; Skorokhodov et al., 2018). An extreme form of leveraging monolingual training data is unsupervised NMT which removes the need for parallel training data entirely (Lample et al., 2017; Artetxe et al., 2017b).

11. NMT Training

NMT models are normally trained using backpropagation (Rumelhart et al., 1988) and a gradient-based optimizer like Adadelta (Zeiler, 2012) with cross-entropy loss. Modern NMT architectures like the Transformer, ConvS2S, or recurrent networks with LSTM or GRU cells help to address known training problems like vanishing gradients (Hochreiter et al., 2001).

However, there is evidence that the optimizer still fails to exploit the full potential of NMT models and often gets stuck in suboptima:

1. NMT models vary greatly in performance, even if they use exactly the same architecture, training data, and are trained for the same number of iterations. Sennrich et al. (2016c) observed up to 1 BLEU difference between different models.

2. NMT ensembling (Sec. 15) combines the scores of multiple separately trained NMT models of the same kind. NMT ensembles consistently outperform single NMT by a large margin. The achieved gains through ensembling might indicate difficulties in training of the single models.

Training is therefore still a very active and diverse research topic. We will sketch some of the challenges in this section, but refer to the publication list in Sec. 14 for further insight.

Deep encoders and decoders consisting of multiple layers have now superseded earlier shallow architectures. However, since the gradients have to be back-propagated through more layers, deep architectures – especially recurrent ones – are prone to vanishing gradients (Pascanu et al., 2013) and are thus harder to train. A number of tricks have been proposed recently that make it possible to train deep NMT models reliably. Residual connections (He et al., 2016b) are direct connections that bypass more complex sub-networks in the layer stack. For example, all the architectures presented in Sec. 6.6 (GNMT, ConvS2S, Transformer, RNMT+) add residual connections around attentional, recurrent, or convolutional cells to ease learning (Fig. 13). Another technique to counter vanishing gradients is called *batch normalization* (Ioffe & Szegedy, 2015) which normalizes the hidden activations in each layer in a mini-batch to a mean of zero and a variance of 1. An extension of batch normalization which is independent of the batch size and is especially suitable for recurrent networks is called *layer normalization* (Ba et al., 2016). Layer normalization is popular for training deep NLP models like the Transformer.

11.1 Regularization

Modern NMT architectures are vastly over-parameterized (Stahlberg & Byrne, 2017) to help training (Livni et al., 2014). For example, a subword-unit-level Transformer in a standard "big" configuration can easily have 200-300 million parameters (Stahlberg et al., 2018b). The large number of parameters potentially makes the model prone to *over-fitting*: The model fits the training data perfectly, but the performance on held-out data suffers as the large number of parameters allows the optimizer to marginally improve training loss at the cost of generalization as training proceeds. Techniques that aim to prevent over-fitting in over-parameterized neural networks are called *regularizers*. Perhaps the two simplest regularization techniques are L1 and L2 regularization. The idea is to add terms to the loss function that penalize the magnitude of weights in the network. Intuitively, such penalties draw many parameters towards zero and limit their significance. Thus, L1 and L2 effectively serve as soft constraints on the model capacity.

The three most popular regularization techniques for NMT are *early stopping*, *dropout*, and *label smoothing*. Early stopping can be seen as regularization in time as it stops training as soon as the performance on the development set does not improve anymore. Dropout (Srivastava et al., 2014) is arguably one of the key techniques that have made deep learning

practical. Dropout randomly sets the activities of hidden and visible units to zero during training. Thus, it can be seen as a strong regularizer for simultaneously training a large collection of networks with extensive weight sharing. Label smoothing has been derived for expectation–maximization training by Byrne (1993), and has been applied to large-scale computer vision by Szegedy et al. (2016). Label smoothing changes the training objective such that the model produces smoother distributions. Other popular methods to mitigate over-fitting include gradient and weight noise, gradient clipping/scaling, learning rate schedules, and adding input noise (e.g. by masking, swapping, or dropping input tokens).

12. Explainable Neural Machine Translation

Explaining the predictions of deep neural models is hard because they consist of tens of thousands of neurons and millions of parameters. Therefore, explainable and interpretable deep learning is still an open research question (Ribeiro et al., 2016; Doshi-Velez & Kim, 2017; Lipton, 2018; Montavon et al., 2018; Alishahi et al., 2019).

12.1 Post-Hoc Interpretability

Post-hoc interpretability refers to the idea of sidestepping the model complexity by treating it as a black-box and not trying to understand the inner workings of the model. Montavon et al. (2018) defines post-hoc interpretability as follows: "A trained model is given and our goal is to understand what the model predicts (e.g. categories) in terms what is readily interpretable (e.g. the input variables)". In NMT, this means that we try to understand the target tokens ("what the model predicts") in terms of the source tokens ("the input variables"). Post-hoc intepretability methods such as layer-wise relevance propagation (Bach et al., 2015) are often visualized with heat maps representing the importance of input variables – pixels in computer vision or source words in machine translation.

Applying post-hoc interpretability methods to sequence-to-sequence prediction has received some attention in the literature (Schwarzenberg et al., 2019). Alvarez-Melis and Jaakkola (2017) proposed a causal model which finds related source-target pairs by feeding in perturbed versions of the source sentence. Ma et al. (2018) derived relevance scores for NMT by comparing the predictive probability distributions before and after zeroing out a particular source word. Feng et al. (2018) point out some general limitations of such post-hoc analyses in NLP.

12.2 Model-Intrinsic Interpretability

Unlike the black-box methods for post-hoc interpretability, another line of research tries to understand the functions of individual hidden neurons or layers in the NMT network. Different methods have been proposed to visualize the activities or gradients in hidden layers (Karpathy et al., 2015; Li et al., 2016a; Ding et al., 2017; Cashman et al., 2018). Belinkov et al. (2017) shed some light on NMT's ability to handle morphology by investigating how well a classifier can predict part-of-speech or morphological tags from the last encoder hidden layer. Bau et al. (2018), Dalvi et al. (2018, 2019) found individual neurons that capture certain linguistic properties with different forms of regression analysis. Bau et al. (2018) were even able to alter the translation (e.g. change the gender) by manipulating

the activities in these neurons. Other researchers have focused on the attention layer. Tang et al. (2018b) suggested that attention at different layers of the Transformer serves different purposes. They also showed that NMT does not use the means of attention for word sense disambiguation. Ghader and Monz (2017) provide a detailed analysis of how NMT uses attention to condition on the source sentence.

12.3 Confidence Estimation in Translation

Obtaining word level or sentence level confidence scores for translations is not only very useful for practical MT, it also improves the explainability and trustworthiness of the MT system. An obvious candidate for confidence scores from an NMT system are the probabilities the model assigns to tokens or sentences. However, there is some disagreement in the literature on how well NMT models are calibrated (Ott et al., 2018; Kumar & Sarawagi, 2019). Poorly calibrated models do not assign probabilities according to the true data distribution. Such models might still assign high scores to high quality translations, but their output distributions are not a reliable source for deriving word-level confidence scores. While confidence estimation has been explored for traditional SMT (de Gispert et al., 2013; Bach et al., 2011; Ueffing & Ney, 2005), it has received almost no attention since the advent of neural machine translation. The only work on confidence in NMT we are aware of is from Rikters and Fishel (2017) and Rikters (2018) who aim to use attention to estimate word-level confidences.

In contrast, the related field of Quality Estimation for MT enjoys great popularity, with well-attended annual WMT evaluation campaigns – by now in their seventh edition (Specia et al., 2018). Quality estimation aims to find meaningful quality metrics which are more accepted by users and customers than abstract metrics like BLEU (Papineni et al., 2002), and are more correlated to the usefulness of MT in a real-world scenario. Possible applications for quality estimation include estimating post-editing efficiency (Specia, 2011) or selecting sentences in the MT output which need human revision (Bach et al., 2011).

13. NMT-SMT Hybrid Systems

Neural models were increasingly used as features in traditional SMT until NMT evolved as a new paradigm. Without question, NMT has become the prevalent approach to machine translation in recent years. There is a large body of research comparing NMT and SMT (Tab. 6). Most studies have found superior overall translation quality of NMT models in most settings, but complementary strengths of both paradigms. Therefore, the literature about hybrid NMT-SMT systems is also vast. We distinguish between two categories of approaches for blending SMT and NMT.

Approaches in the first category do not employ a full SMT system but borrow only key ideas or components from SMT to address specific issues in NMT. It is straightforward to combine NMT scores with other features normally used in SMT (like language models) in a log-linear model (Gulcehre et al., 2015; He et al., 2016c).[14] Conventional symbolic SMT-style lexical translation tables can be incorporated into the NMT decoder by using the soft

14. Note that this is still different from using neural features in an SMT system as the standard left-to-right NMT decoder is used.

Neural machine translation	Statistical machine translation
+ Much better overall translation quality than SMT with enough training data (Koehn & Knowles, 2017; Toral & Sánchez-Cartagena, 2017; Bentivogli et al., 2016, 2018; Castilho et al., 2017b; Junczys-Dowmunt et al., 2016a; Volkart et al., 2018).	+ Outperforms NMT in low-resource scenarios (Koehn & Knowles, 2017; Menacer et al., 2017; Dowling et al., 2018; Jauregi Unanue et al., 2018; Mahata et al., 2018; Ojha et al., 2018).
+ More fluent than SMT (Bentivogli et al., 2016; Toral & Sánchez-Cartagena, 2017; Castilho et al., 2017b; Mahata et al., 2018; Castilho et al., 2017a).	+ Produces richer output lattices (Stahlberg et al., 2016).
+ Better handles a variety of linguistic phenomena than SMT (Bentivogli et al., 2016, 2018; Isabelle et al., 2017).	+ More robust against noise (Ruiz et al., 2017; Khayrallah & Koehn, 2018).
− Adequacy issues due to lack of explicit coverage mechanism (Tu et al., 2016; Yang et al., 2018a; Kong et al., 2018; Mahata et al., 2018; Castilho et al., 2017a).	+ Translation quality degrades less on very long sentences than NMT (Toral & Sánchez-Cartagena, 2017; Bentivogli et al., 2016).
− Lack of hypothesis diversity (Sec. 7.6).	+ Less errors in the translation of proper nouns (Bentivogli et al., 2018).
− Neural models perform not as well as specialized symbolic models on several monotone seq2seq tasks (Schnober et al., 2016).	∘ NMT and SMT require comparable amounts of (document-level) post-editing (Jia et al., 2019; Castilho et al., 2017b).

Table 6: Summary of studies comparing traditional statistical machine translation and neural machine translation.

alignment weights of the standard NMT attention model (He et al., 2016c; Arthur et al., 2016; Zhang & Zong, 2016a; Neubig, 2016; Tang et al., 2016). Cohn et al. (2016) proposed to enhance the attention model in NMT by implementing basic concepts from the original word alignment models (Brown et al., 1993; Vogel et al., 1996) like fertility and relative distortion.

The second category of hybrid systems is related to system combination. The idea is to combine a fully trained SMT system with an independently trained NMT system. Popular examples in this category are rescoring and reranking methods (Neubig et al., 2015; Stahlberg et al., 2016; Khayrallah et al., 2017; Grundkiewicz & Junczys-Dowmunt, 2018; Avramidis et al., 2016; Marie & Fujita, 2018; Zhang et al., 2017), although these models may be too constraining if the neural system is much stronger. Stahlberg et al. (2016) proposed a finite state transducer based loose combination scheme that combines NMT and SMT translations via an edit distance based loss. The minimum Bayes risk (MBR) based approach of Stahlberg et al. (2017) biases an unconstrained NMT decoder towards n-grams which are likely according to the SMT system, and therefore also does not constrain the system to the SMT search space. MBR-based combination of NMT and SMT has been used in WMT evaluation systems (Stahlberg et al., 2018b, 2019) and in industry (Iglesias

et al., 2018). NMT and SMT can also be combined in a cascade, with SMT providing the input to a post-processing NMT system (Niehues et al., 2016; Zhou et al., 2017) or vice versa (Du & Way, 2017). Wang et al. (2017, 2018a) interpolated NMT posteriors with word recommendations from SMT and jointly trained NMT together with a gating function which assigns the weight between SMT and NMT scores dynamically. The AMU-UEDIN submission to WMT16 let SMT take the lead and used NMT as a feature in phrase-based MT (Junczys-Dowmunt et al., 2016b). In contrast, Long et al. (2016) translated most of the sentence with an NMT system, and just used SMT to translate technical terms in a post-processing step. Dahlmann et al. (2017) proposed a hybrid search algorithm in which the neural decoder expands hypotheses with phrases from an SMT system. SMT can also be used as regularizer in unsupervised NMT (Ren et al., 2019).

14. Further Reading

A number of current research efforts are not covered by this article. The following list provides initial reading suggestions for advanced topics in NMT.

- Multimodal NMT (Elliott et al., 2015; Hitschler et al., 2016; Barrault et al., 2018; Calixto & Liu, 2019)
- Tree- or lattice-based NMT (Currey & Heafield, 2018; Aharoni & Goldberg, 2017; Saunders et al., 2018; Nadejde et al., 2017; Sperber et al., 2017; Su et al., 2017),
- Factored NMT (Koehn & Hoang, 2007; Sennrich & Haddow, 2016; García-Martínez et al., 2016, 2017)
- Document-level context (Bawden et al., 2018; Läubli et al., 2018; Müller et al., 2018; Bojar et al., 2019; Yu et al., 2020)
- NMT model shrinking and reduced precision (Wu et al., 2016; See et al., 2016; Kim & Rush, 2016; Freitag et al., 2017; Zhang et al., 2018)
- Multilingual NMT (Johnson et al., 2017; Dabre et al., 2019; Aharoni et al., 2019)
- Low-resource NMT (Koehn & Knowles, 2017; Tong et al., 2018; Bojar et al., 2019, 2018, 2017)
- Unsupervised NMT (Conneau et al., 2017; Artetxe et al., 2017a; Hoshen & Wolf, 2018; Lample et al., 2017; Artetxe et al., 2017b)
- Domain adaptation (Chu & Wang, 2018; Chu et al., 2018; Luong & Manning, 2015; Thompson et al., 2019; Saunders et al., 2019)
- Data filtering (Resnik, 1999; Khayrallah & Koehn, 2018; Carpuat et al., 2017; Junczys-Dowmunt, 2018a; Rossenbach et al., 2018; Junczys-Dowmunt, 2018b)
- Word alignments (Mi et al., 2016b; Alkhouli & Ney, 2017; Alkhouli et al., 2016; Zenkel et al., 2019; Alkhouli et al., 2018; Stahlberg et al., 2018)

- Various extensions to the Transformer architecture (Shaw et al., 2018; Ahmed et al., 2017; Guo et al., 2019; Medina & Kalita, 2018)
- Memory-augmented NMT (Wang et al., 2016; Feng et al., 2017; Li et al., 2019; Xiong et al., 2018)
- Variational methods (Zhang et al., 2016; Su et al., 2018; Bastings et al., 2019; Shah & Barber, 2018)
- Non- or partially autoregressive architectures (Wang et al., 2018a; Gu et al., 2017a; Guo et al., 2018; Wang et al., 2019; Libovický & Helcl, 2018; Lee et al., 2018; Akoury et al., 2019)
- Simultaneous translation (Lewis, 2015; Mieno et al., 2015; Cho & Esipova, 2016; Gu et al., 2017)
- Large batch training (Popel & Bojar, 2018; McCandlish et al., 2018; Stahlberg et al., 2018b; Saunders et al., 2018; Neishi et al., 2017; Morishita et al., 2017)
- Reinforcement learning and risk-based training (Ranzato et al., 2015; Wu et al., 2016, 2018; Shen et al., 2016; Edunov et al., 2018b)
- Adversarial training (Zhang et al., 2018b; Yang et al., 2018; Wu et al., 2017).

For even more insight into the field of neural machine translation, we refer the reader to other overview papers from Neubig (2017), Cromieres et al. (2017), Koehn (2017), Popescu-Belis (2019).

15. Conclusion

Neural machine translation (NMT) has become the de facto standard for large-scale machine translation in a very short period of time. This article traced back the origin of NMT to word and sentence embeddings and neural language models. We reviewed the most commonly used building blocks of NMT architectures – recurrence, convolution, and attention – and discussed popular concrete architectures such as RNNsearch, GNMT, ConvS2S, and the Transformer. We discussed the advantages and disadvantages of several important design choices that have to be made to design a good NMT system with respect to decoding, training, and segmentation. We then shortly explored advanced topics in NMT research such as explainability and NMT-SMT hybrid systems.

Acknowledgments

I am grateful to Bill Byrne for his comments on an earlier version of the manuscript, and to the associate editor Stephen Clark whose advice was vital for shaping this review article. I also thank the surveys editor Dragomir Radev for his guidance through the entire publication process, and all the anonymous reviewers for their detailed feedback. This article draws from the author's PhD thesis (EPSRC grant EP/L027623/1 and EPSRC Tier-2 grant EP/P020259/1), but was finalized while the author was at Google Research.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, Savannah, GA. USENIX Association.

Adel, H., & Schütze, H. (2017). Exploring different dimensions of attention for uncertainty detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 22–34, Valencia, Spain. Association for Computational Linguistics.

Aharoni, R., & Goldberg, Y. (2017). Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 132–140, Vancouver, Canada. Association for Computational Linguistics.

Aharoni, R., Johnson, M., & Firat, O. (2019). Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Ahmed, K., Keskar, N. S., & Socher, R. (2017). Weighted Transformer network for machine translation. *arXiv preprint arXiv:1711.02132*.

Akoury, N., Krishna, K., & Iyyer, M. (2019). Syntactically supervised Transformers for faster neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy. Association for Computational Linguistics.

Alishahi, A., Chrupala, G., & Linzen, T. (2019). Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Natural Language Engineering*.

Alkhouli, T., Bretschner, G., & Ney, H. (2018). On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 177–185, Belgium, Brussels. Association for Computational Linguistics.

Alkhouli, T., Bretschner, G., Peter, J.-T., Hethnawi, M., Guta, A., & Ney, H. (2016). Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pp. 54–65, Berlin, Germany. Association for Computational Linguistics.

Alkhouli, T., & Ney, H. (2017). Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pp. 108–117, Copenhagen, Denmark. Association for Computational Linguistics.

Alvarez-Melis, D., & Jaakkola, T. (2017). A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on*

Empirical Methods in Natural Language Processing, pp. 412–421, Copenhagen, Denmark. Association for Computational Linguistics.

Álvaro Peris, & Casacuberta, F. (2018). NMT-Keras: A very flexible toolkit with a focus on interactive NMT and online learning. *The Prague Bulletin of Mathematical Linguistics*, *111*, 113–124.

Andreas, J., & Klein, D. (2015). When and why are log-linear models self-normalizing?. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 244–249, Denver, Colorado. Association for Computational Linguistics.

Artetxe, M., Labaka, G., & Agirre, E. (2017a). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 451–462, Vancouver, Canada. Association for Computational Linguistics.

Artetxe, M., Labaka, G., Agirre, E., & Cho, K. (2017b). Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Arthur, P., Neubig, G., & Nakamura, S. (2016). Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1557–1567, Austin, Texas. Association for Computational Linguistics.

Ataman, D., Negri, M., Turchi, M., & Federico, M. (2017). Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *The Prague Bulletin of Mathematical Linguistics*, *108*(1), 331–342.

Avramidis, E., Macketanz, V., Burchardt, A., Helcl, J., & Uszkoreit, H. (2016). Deeper machine translation and evaluation for German. In *Proceedings of the 2nd Deep Machine Translation Workshop*, pp. 29–38, Lisbon, Portugal. ÚFAL MFF UK.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Ba, J. L., Mnih, V., & Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.

Bach, N., Huang, F., & Al-Onaizan, Y. (2011). Goodness: A method for measuring machine translation confidence. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 211–219, Portland, Oregon, USA. Association for Computational Linguistics.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, *10*(7), 1–46.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*.

Barrault, L., Bougares, F., Specia, L., Lala, C., Elliott, D., & Frank, S. (2018). Findings of the third shared task on multimodal machine translation. In *Proceedings of the*

Third Conference on Machine Translation: Shared Task Papers, pp. 304–323, Belgium, Brussels. Association for Computational Linguistics.

Basho, & Reichhold, J. (2013). *Basho: the complete haiku*. Kodansha International.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., & Bengio, Y. (2012). Theano: New features and speed improvements. In *NIPS*.

Bastings, J., Aziz, W., Titov, I., & Sima'an, K. (2019). Modeling latent sentence structure in neural machine translation. *arXiv preprint arXiv:1901.06436*.

Bau, A., Belinkov, Y., Sajjad, H., Durrani, N., Dalvi, F., & Glass, J. (2018). Identifying and controlling important neurons in neural machine translation. *arXiv preprint arXiv:1811.01157*.

Bawden, R., Sennrich, R., Birch, A., & Haddow, B. (2018). Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Belinkov, Y., & Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., & Glass, J. (2017). What do neural machine translation models learn about morphology?. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 861–872, Vancouver, Canada. Association for Computational Linguistics.

Bellegarda, J. R. (1997). A latent semantic analysis framework for large-span language modeling. In *Eurospeech*.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137–1155.

Bengio, Y., Mesnil, G., Dauphin, Y. N., & Rifai, S. (2013). Better mixing via deep representations. In Dasgupta, S., & McAllester, D. (Eds.), *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28 of *Proceedings of Machine Learning Research*, pp. 552–560, Atlanta, Georgia, USA. PMLR.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). *Neural Probabilistic Language Models*, pp. 137–186. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bentivogli, L., Bisazza, A., Cettolo, M., & Federico, M. (2016). Neural versus phrase-based machine translation quality: A case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 257–267, Austin, Texas. Association for Computational Linguistics.

Bentivogli, L., Bisazza, A., Cettolo, M., & Federico, M. (2018). Neural versus phrase-based mt quality: An in-depth analysis on English–German and English–French. *Computer Speech & Language*, *49*, 52 – 70.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia,

L., & Turchi, M. (2017). Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pp. 169–214. Association for Computational Linguistics.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., & Zampieri, M. (2016). Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pp. 131–198, Berlin, Germany. Association for Computational Linguistics.

Bojar, O., et al. (2019). Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation: Shared Task Papers*. Association for Computational Linguistics.

Bojar, O., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Koehn, P., & Monz, C. (2018). Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 272–303. Association for Computational Linguistics.

Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2013). Audio chord recognition with recurrent neural networks.. In *ISMIR*, pp. 335–340. Citeseer.

Bowman, S., Pavlick, E., Grave, E., van Durme, B., Wang, A., Hula, J., Xia, P., Pappagari, R., McCoy, R. T., Patel, R., et al. (2018). Looking for ELMo's friends: Sentence-level pretraining beyond language modeling. *arXiv preprint arXiv:1812.10860*.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, *19*(2), 263–311.

Buciluǎ, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 535–541, New York, NY, USA. ACM.

Burlot, F., & Yvon, F. (2018). Using monolingual data in neural machine translation: A systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 144–155, Belgium, Brussels. Association for Computational Linguistics.

Byrne, B. (1993). Generalization and maximum likelihood from small data sets. In *Neural Networks for Signal Processing III - Proceedings of the 1993 IEEE-SP Workshop*, pp. 197–206.

Calixto, I., & Liu, Q. (2019). An error analysis for image-based multi-modal neural machine translation. *Machine Translation*.

Carpuat, M., Vyas, Y., & Niu, X. (2017). Detecting cross-lingual semantic divergence for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 69–79, Vancouver. Association for Computational Linguistics.

Cashman, D., Patterson, G., Mosca, A., Watts, N., Robinson, S., & Chang, R. (2018). RNNbow: Visualizing learning via backpropagation gradients in RNNs. *IEEE Computer Graphics and Applications*, *38*(6), 39–50.

Castilho, S., Moorkens, J., Gaspari, F., Calixto, I., Tinsley, J., & Way, A. (2017a). Is neural machine translation the new state of the art?. *The Prague Bulletin of Mathematical Linguistics*, *108*(1), 109–120.

Castilho, S., Moorkens, J., Gaspari, F., Sennrich, R., Sosoni, V., Georgakopoulou, P., Lohar, P., Way, A., Barone, A. V. M., & Gialama, M. (2017b). A comparative quality evaluation of PBSMT and NMT using professional translators. *Proceedings of Machine Translation Summit XVI, Nagoya, Japan.*

Chan, W., Jaitly, N., Le, Q. V., & Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964.

Chen, M. X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Schuster, M., Shazeer, N., Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Chen, Z., Wu, Y., & Hughes, M. (2018a). The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 76–86, Melbourne, Australia. Association for Computational Linguistics.

Chen, Y., Li, V. O., Cho, K., & Bowman, S. (2018b). A stable and effective learning strategy for trainable greedy decoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 380–390, Brussels, Belgium. Association for Computational Linguistics.

Cheng, J., Dong, L., & Lapata, M. (2016a). Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 551–561, Austin, Texas. Association for Computational Linguistics.

Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., & Liu, Y. (2016b). Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1965–1974, Berlin, Germany. Association for Computational Linguistics.

Cherry, C., Foster, G., Bapna, A., Firat, O., & Macherey, W. (2018). Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *American Journal of Computational Linguistics*, *33*(2), 201–228.

Chitnis, R., & DeNero, J. (2015). Variable-length word encodings for neural translation models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2088–2093, Lisbon, Portugal. Association for Computational Linguistics.

Cho, K. (2016). Noisy parallel approximate decoding for conditional recurrent language model. *arXiv preprint arXiv:1605.03835.*

Cho, K., & Esipova, M. (2016). Can neural machine translation do simultaneous translation?. *arXiv preprint arXiv:1606.02012*.

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014a). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Doha, Qatar. Association for Computational Linguistics.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Choi, H., Cho, K., & Bengio, Y. (2017). Context-dependent word representation for neural machine translation. *Computer Speech & Language*, *45*, 149 – 160.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807. IEEE.

Chorowski, J., Bahdanau, D., Cho, K., & Bengio, Y. (2014). End-to-end continuous speech recognition using attention-based recurrent NN: First results. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Chu, C., Dabre, R., & Kurohashi, S. (2018). A comprehensive empirical comparison of domain adaptation methods for neural machine translation. *Journal of Information Processing*, *26*, 529–538.

Chu, C., & Wang, R. (2018). A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chung, J., Cho, K., & Bengio, Y. (2016). A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Cohn, T., Hoang, C. D. V., Vymolova, E., Yao, K., Dyer, C., & Haffari, G. (2016). Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 876–885, San Diego, California. Association for Computational Linguistics.

Collobert, R., Hannun, A., & Synnaeve, G. (2019). A fully differentiable beam search decoder. *arXiv preprint arXiv:1902.06022*.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 160–167, New York, NY, USA. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, *12*, 2493–2537.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 670–680. Association for Computational Linguistics.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136. Association for Computational Linguistics.

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., & Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Costa-jussà, M. R., Escolano, C., & Fonollosa, J. A. (2017). Byte-based neural machine translation. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pp. 154–158, Copenhagen, Denmark. Association for Computational Linguistics.

Costa-jussà, M. R., & Fonollosa, J. A. (2016). Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 357–361, Berlin, Germany. Association for Computational Linguistics.

Crego, J., Kim, J., Klein, G., Rebollo, A., Yang, K., Senellart, J., Akhanov, E., Brunelle, P., Coquard, A., Deng, Y., et al. (2016). SYSTRAN's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Cromieres, F., Chu, C., Nakazawa, T., & Kurohashi, S. (2016). Kyoto university participation to WAT 2016. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pp. 166–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Cromieres, F., Nakazawa, T., & Dabre, R. (2017). Neural machine translation: Basics, practical aspects and recent trends. In *Proceedings of the IJCNLP 2017, Tutorial Abstracts*, pp. 11–13, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Currey, A., & Heafield, K. (2018). Multi-source syntactic neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2961–2966, Brussels, Belgium. Association for Computational Linguistics.

Currey, A., Miceli Barone, A. V., & Heafield, K. (2017). Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pp. 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

Dabre, R., Chu, C., & Kunchukuttan, A. (2019). A survey of multilingual neural machine translation. *arXiv preprint arXiv:1905.05395*.

Dahlmann, L., Matusov, E., Petrushkov, P., & Khadivi, S. (2017). Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1411–1420, Copenhagen, Denmark. Association for Computational Linguistics.

Dalvi, F., Durrani, N., Sajjad, H., Belinkov, Y., Bau, A., & Glass, J. (2019). What is one grain of sand in the desert? Analyzing individual neurons in deep NLP models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Dalvi, F., Nortonsmith, A., Bau, A., Belinkov, Y., Sajjad, H., Durrani, N., & Glass, J. (2018). NeuroX: A toolkit for analyzing individual neurons in neural networks. *arXiv preprint arXiv:1812.09359*.

Daniil, G., Kalaidin, P., & Malykh, V. (2019). Self-attentive model for headline generation. *arXiv preprint arXiv:1901.07786*.

de Gispert, A., Blackwood, G., Iglesias, G., & Byrne, B. (2013). N-gram posterior probability confidence measures for statistical machine translation: An empirical study. *Machine Translation*, *27*(2), 85–114.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., & Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1370–1380, Baltimore, Maryland. Association for Computational Linguistics.

Di Gangi, M. A., & Federico, M. (2018). Deep neural machine translation with weakly-recurrent units. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation: 28-30 May 2018, Universitat d'Alacant, Alacant, Spain*, pp. 119–128. European Association for Machine Translation.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ding, Y., Liu, Y., Luan, H., & Sun, M. (2017). Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1150–1159, Vancouver, Canada. Association for Computational Linguistics.

Domhan, T. (2018). How much attention do you need? A granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1799–1808, Melbourne, Australia. Association for Computational Linguistics.

Domhan, T., & Hieber, F. (2017). Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.

Domingo, M., Garcıa-Martınez, M., Helle, A., & Casacuberta, F. (2018). How much does tokenization affect in neural machine translation?. *arXiv preprint arXiv:1812.08621*.

Dong, L., & Lapata, M. (2016). Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33–43, Berlin, Germany. Association for Computational Linguistics.

dos Santos, C., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78. Dublin City University and Association for Computational Linguistics.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Dowling, M., Lynn, T., Poncelas, A., & Way, A. (2018). SMT versus NMT: Preliminary comparisons for irish. In *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)*, pp. 12–20, Boston, MA. Association for Machine Translation in the Americas.

Du, J., & Way, A. (2017). Neural pre-translation for hybrid machine translation. In *Proceedings of MT Summit*, Vol. 16, pp. 27–40.

Duong, L., Anastasopoulos, A., Chiang, D., Bird, S., & Cohn, T. (2016). An attentional model for speech translation without transcription. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 949–959, San Diego, California. Association for Computational Linguistics.

Durrani, N., Dalvi, F., Sajjad, H., Belinkov, Y., & Nakov, P. (2018). What is in a translation unit? Comparing character and subword representations beyond translation. *openreview.net*.

Durrani, N., Dalvi, F., Sajjad, H., & Vogel, S. (2016). QCRI machine translation systems for IWSLT 16. In *International Workshop on Spoken Language Translation. Seattle, WA, USA*.

Dyer, C. (2014). Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*.

Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018a). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, Brussels, Belgium. Association for Computational Linguistics.

Edunov, S., Ott, M., Auli, M., Grangier, D., & Ranzato, M. (2018b). Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Elliott, D., Frank, S., & Hasler, E. (2015). Multilingual image description with neural sequence models. *arXiv preprint arXiv:1510.04709*.

Er, M. J., Zhang, Y., Wang, N., & Pratama, M. (2016). Attention pooling-based convolutional neural network for sentence modelling. *Information Sciences*, *373*, 388 – 403.

Escolano, C., Costa-jussà, M. R., & Fonollosa, J. A. (2018). (self-attentive) autoencoder-based universal language representation for machine translation. *arXiv preprint arXiv:1810.06351*.

Fan, A., Grangier, D., & Auli, M. (2018). Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 45–54, Melbourne, Australia. Association for Computational Linguistics.

Feng, S., Wallace, E., Grissom II, A., Iyyer, M., Rodriguez, P., & Boyd-Graber, J. (2018). Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Feng, Y., Zhang, S., Zhang, A., Wang, D., & Abel, A. (2017). Memory-augmented neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1390–1399, Copenhagen, Denmark. Association for Computational Linguistics.

Freitag, M., & Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Freitag, M., & Al-Onaizan, Y. (2017). Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 56–60, Vancouver. Association for Computational Linguistics.

Freitag, M., Al-Onaizan, Y., & Sankaran, B. (2017). Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, *12*(2), 23–38.

García-Martínez, M., Barrault, L., & Bougares, F. (2016). Factored neural machine translation architectures. In *International Workshop on Spoken Language Translation (IWSLT'16)*.

García-Martínez, M., Barrault, L., & Bougares, F. (2017). Neural machine translation by generating multiple linguistic factors. In Camelin, N., Estève, Y., & Martín-Vide, C. (Eds.), *Statistical Language and Speech Processing*, pp. 21–31, Cham. Springer International Publishing.

Gehring, J., Auli, M., Grangier, D., & Dauphin, Y. N. (2017a). A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 123–135, Vancouver, Canada. Association for Computational Linguistics.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017b). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1243–1252. JMLR.org.

Ghader, H., & Monz, C. (2017). What does attention in neural machine translation pay attention to?. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 30–39, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Gimpel, K., Batra, D., Dyer, C., & Shakhnarovich, G. (2013). A systematic exploration of diversity in machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1100–1111, Seattle, Washington, USA. Association for Computational Linguistics.

Goel, V., Kumar, S., & Byrne, B. (2000). Segmental minimum Bayes-risk ASR voting strategies. In *Interspeech*, pp. 139–142.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, *57*, 345–420.

Goldberg, Y. (2019). Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Goodfellow, I., Warde-farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks. In *ICML*, pp. 1319–1327.

Goyal, K., Neubig, G., Dyer, C., & Berg-Kirkpatrick, T. (2018). A continuous relaxation of beam search for end-to-end training of neural sequence models. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Grundkiewicz, R., & Junczys-Dowmunt, M. (2018). Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., & Socher, R. (2017a). Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

Gu, J., Cho, K., & Li, V. O. (2017b). Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1968–1978, Copenhagen, Denmark. Association for Computational Linguistics.

Gu, J., Liu, Q., & Cho, K. (2019). Insertion-based decoding with automatically inferred generation order. *arXiv preprint arXiv:1902.01370*.

Gu, J., Neubig, G., Cho, K., & Li, V. O. (2017). Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Gu, J., Wang, C., & Zhao, J. (2019). Levenshtein Transformer. *arXiv preprint arXiv:1905.11006*.

Gulcehre, C., Ahn, S., Nallapati, R., Zhou, B., & Bengio, Y. (2016). Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 140–149, Berlin, Germany. Association for Computational Linguistics.

Gulcehre, C., Dutil, F., Trischler, A., & Bengio, Y. (2017). Plan, attend, generate: Character-level neural machine translation with planning. In *Proceedings of the 2nd Workshop*

on Representation Learning for NLP, pp. 228–234, Vancouver, Canada. Association for Computational Linguistics.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., & Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Gulcehre, C., Firat, O., Xu, K., Cho, K., & Bengio, Y. (2017). On integrating a language model into neural machine translation. *Computer Speech & Language*, *45*, 137 – 148.

Guo, J., Tan, X., He, D., Qin, T., Xu, L., & Liu, T.-Y. (2018). Non-autoregressive neural machine translation with enhanced decoder input. *arXiv preprint arXiv:1812.09664*.

Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., & Zhang, Z. (2019). Star-Transformer. *arXiv preprint arXiv:1902.09113*.

Gutmann, M., & Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Teh, Y. W., & Titterington, M. (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

Hans, K., & Milton, R. (2016). Improving the performance of neural machine translation involving morphologically rich languages. *arXiv preprint arXiv:1612.02482*.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, *12*(10), 993–1001.

Hao, J., Wang, X., Yang, B., Wang, L., Zhang, J., & Tu, Z. (2019). Modeling recurrence for Transformer. *arXiv preprint arXiv:1904.03092*.

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J. H., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W. D., Li, M., et al. (2018). Achieving human parity on automatic Chinese to English news translation. *arXiv preprint arXiv:1803.05567*.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y., & Ma, W.-Y. (2016a). Dual learning for machine translation. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 29*, pp. 820–828. Curran Associates, Inc.

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, W., He, Z., Wu, H., & Wang, H. (2016c). Improved neural machine translation with SMT features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 151–157. AAAI Press.

He, X., Haffari, G., & Norouzi, M. (2018). Sequence to sequence mixture model for diverse machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 583–592, Brussels, Belgium. Association for Computational Linguistics.

Heafield, K., Pouzyrevsky, I., Clark, J. H., & Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the*

Association for Computational Linguistics (Volume 2: Short Papers), pp. 690–696, Sofia, Bulgaria. Association for Computational Linguistics.

Helcl, J., & Libovický, J. (2017). Neural Monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, 5–17.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 28*, pp. 1693–1701. Curran Associates, Inc.

Hieber, F., Domhan, T., Denkowski, M., Vilar, D., Sokolov, A., Clifton, A., & Post, M. (2017). Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Hitschler, J., Schamoni, S., & Riezler, S. (2016). Multimodal pivots for image caption translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2399–2409, Berlin, Germany. Association for Computational Linguistics.

Hoang, V. C. D., Koehn, P., Haffari, G., & Cohn, T. (2018). Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 18–24, Melbourne, Australia. Association for Computational Linguistics.

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: The difficulty of learning long-term dependencies..

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780.

Hoshen, Y., & Wolf, L. (2018). Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 469–478, Brussels, Belgium. Association for Computational Linguistics.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, M., Peng, Y., Huang, Z., Qiu, X., Wei, F., & Zhou, M. (2018). Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 4099–4106. AAAI Press.

Huang, L., Zhao, K., & Ma, M. (2017). When to finish? Optimal beam search for neural text generation (modulo beam size). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2134–2139, Copenhagen, Denmark. Association for Computational Linguistics.

Huck, M., Riess, S., & Fraser, A. (2017). Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pp. 56–67, Copenhagen, Denmark. Association for Computational Linguistics.

Iglesias, G., Tambellini, W., de Gispert, A., Hasler, E., & Byrne, B. (2018). Accelerating NMT batched beam decoding with LMBR posteriors for deployment. In *Proceedings*

of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pp. 106–113, New Orleans - Louisiana. Association for Computational Linguistics.

Im, J., & Cho, S. (2017). Distance-based self-attention network for natural language inference. *arXiv preprint arXiv:1712.02047*.

Imamura, K., Fujita, A., & Sumita, E. (2018). Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 55–63, Melbourne, Australia. Association for Computational Linguistics.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org.

Isabelle, P., Cherry, C., & Foster, G. (2017). A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Jauregi Unanue, I., Garmendia Arratibel, L., Zare Borzeshi, E., & Piccardi, M. (2018). English-Basque statistical and neural machine translation. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Jean, S., Cho, K., Memisevic, R., & Bengio, Y. (2015a). On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1–10, Beijing, China. Association for Computational Linguistics.

Jean, S., Firat, O., Cho, K., Memisevic, R., & Bengio, Y. (2015b). Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 134–140, Lisbon, Portugal. Association for Computational Linguistics.

Jia, Y., Carl, M., & Wang, X. (2019). Post-editing neural machine translation versus phrase-based machine translation for English–Chinese. *Machine Translation*, 1–21.

Johansen, A. R., Hansen, J. M., Obeid, E. K., Sønderby, C. K., & Winther, O. (2016). Neural machine translation with characters and hierarchical encoding. *arXiv preprint arXiv:1610.06550*.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., & Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, *5*, 339–351.

Junczys-Dowmunt, M. (2018a). Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 888–895, Belgium, Brussels. Association for Computational Linguistics.

Junczys-Dowmunt, M. (2018b). Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 425–430, Belgium, Brussels. Association for Computational Linguistics.

Junczys-Dowmunt, M., Dwojak, T., & Hoang, H. (2016a). Is neural machine translation ready for deployment? A case study on 30 translation directions. In *International Workshop on Spoken Language Translation IWSLT*.

Junczys-Dowmunt, M., Dwojak, T., & Sennrich, R. (2016b). The AMU-UEDIN submission to the WMT16 news translation task: Attention-based NMT models as feature functions in phrase-based SMT. In *Proceedings of the First Conference on Machine Translation*, pp. 319–325, Berlin, Germany. Association for Computational Linguistics.

Kaiser, Ł., Gomez, A. N., & Chollet, F. (2017). Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*.

Kaiser, Ł., Roy, A., Vaswani, A., Pamar, N., Bengio, S., Uszkoreit, J., & Shazeer, N. (2018). Fast decoding in sequence models using discrete latent variables. *arXiv preprint arXiv:1803.03382*.

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Kalchbrenner, N., Espeholt, L., Simonyan, K., van den Oord, A., Graves, A., & Kavukcuoglu, K. (2016). Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 655–665. Association for Computational Linguistics.

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Kazimi, M. B., & Costa-Jussá, M. R. (2017). Coverage for character based neural machine translation. *Procesamiento del Lenguaje Natural*, *59*(0), 99–106.

Khayrallah, H., & Koehn, P. (2018). On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 74–83, Melbourne, Australia. Association for Computational Linguistics.

Khayrallah, H., Kumar, G., Duh, K., Post, M., & Koehn, P. (2017). Neural lattice search for domain adaptation in machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 20–25, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., & Okumura, M. (2016). Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1328–1338, Austin, Texas. Association for Computational Linguistics.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751. Association for Computational Linguistics.

Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2016). Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2741–2749. AAAI Press.

Kim, Y., & Rush, A. M. (2016). Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas. Association for Computational Linguistics.

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pp. 67–72, Vancouver, Canada. Association for Computational Linguistics.

Koehn, P. (2004). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In Frederking, R. E., & Taylor, K. B. (Eds.), *Machine Translation: From Real Users to Research*, pp. 115–124, Berlin, Heidelberg. Springer Berlin Heidelberg.

Koehn, P. (2010). *Statistical Machine Translation* (1st edition). Cambridge University Press, New York, NY, USA.

Koehn, P. (2017). Neural machine translation. *arXiv preprint arXiv:1709.07809*.

Koehn, P., & Hoang, H. (2007). Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 868–876, Prague, Czech Republic. Association for Computational Linguistics.

Koehn, P., & Knowles, R. (2017). Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39, Vancouver. Association for Computational Linguistics.

Kong, X., Tu, Z., Shi, S., Hovy, E., & Zhang, T. (2018). Neural machine translation with adequacy-oriented learning. *arXiv preprint arXiv:1811.08541*.

Kreutzer, J., & Sokolov, A. (2018). Optimally segmenting inputs for NMT shows preference for character-level processing. *arXiv preprint arXiv:1810.01480*.

Kuchaiev, O., Ginsburg, B., Gitman, I., Lavrukhin, V., Case, C., & Micikevicius, P. (2018). OpenSeq2Seq: Extensible toolkit for distributed and mixed precision training of sequence-to-sequence models. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pp. 41–46, Melbourne, Australia. Association for Computational Linguistics.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, Melbourne, Australia. Association for Computational Linguistics.

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 66–71, Brussels, Belgium. Association for Computational Linguistics.

Kumar, A., & Sarawagi, S. (2019). Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Kumar, S., & Byrne, B. (2004). Minimum Bayes-risk decoding for statistical machine translation. In *HLT-NAACL 2004: Main Proceedings*, pp. 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Kunchukuttan, A., & Bhattacharyya, P. (2016). Faster decoding for subword level phrase-based SMT between related languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pp. 82–88, Osaka, Japan. The COLING 2016 Organizing Committee.

Kunchukuttan, A., & Bhattacharyya, P. (2017). Learning variable length units for SMT between related languages via byte pair encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pp. 14–24, Copenhagen, Denmark. Association for Computational Linguistics.

Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Larochelle, H., & Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Advances in neural information processing systems*, pp. 1243–1251.

Läubli, S., Sennrich, R., & Volk, M. (2018). Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Le, H.-S., Allauzen, A., & Yvon, F. (2012). Continuous space translation models with neural networks. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 39–48, Montréal, Canada. Association for Computational Linguistics.

Le, Q. V., Luong, M.-T., Sutskever, I., Vinyals, O., & Zaremba, W. (2016). Neural machine translation systems with rare word processing.. US Patent App. 14/921,925.

Lecorvé, G., & Motlicek, P. (2012). Conversion of recurrent neural network language models to weighted finite state transducers for automatic speech recognition. In *Thirteenth Annual Conference of the International Speech Communication Association*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, *1*(4), 541–551.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In Touretzky, D. S. (Ed.), *Advances in Neural Information Processing Systems 2*, pp. 396–404. Morgan-Kaufmann.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lee, C.-Y., & Osindero, S. (2016). Recursive recurrent nets with attention modeling for OCR in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lee, J., Cho, K., & Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, *5*, 365–378.

Lee, J., Mansimov, E., & Cho, K. (2018). Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Levin, P., Dhanuka, N., Khalil, T., Kovalev, F., & Khalilov, M. (2017). Toward a full-scale neural machine translation in production: the booking.com use case. *arXiv preprint arXiv:1709.05820*.

Lewis, W. D. (2015). Skype translator: Breaking down language and hearing barriers. *Translating and the Computer (TC37)*, *10*, 125–149.

L'Hostis, G., Grangier, D., & Auli, M. (2016). Vocabulary selection strategies for neural machine translation. *arXiv preprint arXiv:1610.00072*.

Li, F., Quan, D., Qiang, W., Tong, X., & Zhu, J. (2017). Handling many-to-one unk translation for neural machine translation. In *Machine Translation: 13th China Workshop, CWMT 2017, Revised Selected Papers*, pp. 102–111. Springer.

Li, J., Chen, X., Hovy, E., & Jurafsky, D. (2016a). Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 681–691, San Diego, California. Association for Computational Linguistics.

Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016b). A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California. Association for Computational Linguistics.

Li, J., & Jurafsky, D. (2016). Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*.

Li, P., Liu, Y., & Sun, M. (2013). Recursive autoencoders for ITG-based translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 567–577, Seattle, Washington, USA. Association for Computational Linguistics.

Li, S., Xu, J., Zhang, Y., & Chen, Y. (2017). A method of unknown words processing for neural machine translation using HowNet. In *Machine Translation: 13th China Workshop, CWMT 2017, Revised Selected Papers*, pp. 20–29. Springer.

Li, X., Zhang, J., & Zong, C. (2016). Towards zero unknown word in neural machine translation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 2852–2858. AAAI Press.

Li, Y., Liu, X., Liu, D., Zhang, X., & Liu, J. (2019). Learning efficient lexically-constrained neural machine translation with external memory. *arXiv preprint arXiv:1901.11344*.

Li, Y., Xiao, T., Li, Y., Wang, Q., Xu, C., & Zhu, J. (2018). A simple and effective approach to coverage-aware neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 292–297, Melbourne, Australia. Association for Computational Linguistics.

Libovický, J., & Helcl, J. (2018). End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Lin, J., & Dyer, C. (2010). Data-intensive text processing with MapReduce. In *NAACL HLT 2010 Tutorial Abstracts*, pp. 1–2, Los Angeles, California. Association for Computational Linguistics.

Lin, J., Sun, X., Ren, X., Ma, S., Su, J., & Su, Q. (2018). Deconvolution-based global decoding for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3260–3271, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Ling, W., Trancoso, I., Dyer, C., & Black, A. W. (2015). Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, *16*(3), 30:31–30:57.

Liu, L., Utiyama, M., Finch, A., & Sumita, E. (2016). Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 411–416, San Diego, California. Association for Computational Linguistics.

Liu, N. F., May, J., Pust, M., & Knight, K. (2018). Augmenting statistical machine translation with subword translation of out-of-vocabulary words. *arXiv preprint arXiv:1808.05700*.

Liu, X., Wang, Y., Chen, X., Gales, M. J., & Woodland, P. C. (2014). Efficient lattice rescoring using recurrent neural network language models. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4908–4912.

Liu, Y., Sun, C., Lin, L., & Wang, X. (2016). Learning natural language inference using bidirectional LSTM model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Liu, Y., Luo, Z., & Zhu, K. (2018a). Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4110–4119, Brussels, Belgium. Association for Computational Linguistics.

Liu, Y., Zhou, L., Wang, Y., Zhao, Y., Zhang, J., & Zong, C. (2018b). A comparable study on model averaging, ensembling and reranking in NMT. In Zhang, M., Ng, V., Zhao,

D., Li, S., & Zan, H. (Eds.), *Natural Language Processing and Chinese Computing*, pp. 299–308, Cham. Springer International Publishing.

Livni, R., Shalev-Shwartz, S., & Shamir, O. (2014). On the computational efficiency of training neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 27*, pp. 855–863. Curran Associates, Inc.

Long, Z., Utsuro, T., Mitsuhashi, T., & Yamamoto, M. (2016). Translation of patent sentences with a large vocabulary of technical terms using neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pp. 47–57, Osaka, Japan. The COLING 2016 Organizing Committee.

Luong, M.-T., & Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pp. 76–79.

Luong, M.-T., & Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Luong, M.-T., Pham, H., & Manning, C. D. (2015a). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421. Association for Computational Linguistics.

Luong, M.-T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2015b). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 11–19, Beijing, China. Association for Computational Linguistics.

Ma, X., Li, K., & Koehn, P. (2018). An analysis of source context dependency in neural machine translation. In *21st Annual Conference of the European Association for Machine Translation*, p. 189.

Macháček, D., Vidra, J., & Bojar, O. (2018). Morphological and language-agnostic word segmentation for NMT. In Sojka, P., Horák, A., Kopeček, I., & Pala, K. (Eds.), *Text, Speech, and Dialogue*, pp. 277–284, Cham. Springer International Publishing.

Mahata, S. K., Mandal, S., Das, D., & Bandyopadhyay, S. (2018). SMT vs NMT: a comparison over Hindi & Bengali simple sentences. *arXiv preprint arXiv:1812.04898*.

Malaviya, C., Ferreira, P., & Martins, A. F. T. (2018). Sparse and constrained attention for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 370–376, Melbourne, Australia. Association for Computational Linguistics.

Marie, B., & Fujita, A. (2018). A smorgasbord of features to combine phrase-based and neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pp. 111–124, Boston, MA. Association for Machine Translation in the Americas.

McCandlish, S., Kaplan, J., Amodei, D., & Team, O. D. (2018). An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*.

McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 6294–6305. Curran Associates, Inc.

Medina, J. R., & Kalita, J. (2018). Parallel attention mechanisms in neural machine translation. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 547–552.

Mehri, S., & Sigal, L. (2018). Middle-out decoding. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 31*, pp. 5518–5529. Curran Associates, Inc.

Menacer, M. A., Langlois, D., Mella, O., Fohr, D., Jouvet, D., & Smaïli, K. (2017). Is statistical machine translation approach dead?. In *ICNLSSP 2017 - International Conference on Natural Language, Signal and Speech Processing*, pp. 1–5, Casablanca, Morocco. ISGA.

Mi, H., Sankaran, B., Wang, Z., & Ittycheriah, A. (2016a). Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 955–960, Austin, Texas. Association for Computational Linguistics.

Mi, H., Wang, Z., & Ittycheriah, A. (2016b). Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2283–2288, Austin, Texas. Association for Computational Linguistics.

Mi, H., Wang, Z., & Ittycheriah, A. (2016c). Vocabulary manipulation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 124–129, Berlin, Germany. Association for Computational Linguistics.

Miao, G., Xu, J., Li, Y., Li, S., & Chen, Y. (2017). An unknown word processing method in NMT by integrating syntactic structure and semantic concept. In *Machine Translation: 13th China Workshop, CWMT 2017, Revised Selected Papers*, pp. 43–54. Springer.

Miculicich, L., Pappas, N., Ram, D., & Popescu-Belis, A. (2018). Self-attentive residual decoder for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1366–1379, New Orleans, Louisiana. Association for Computational Linguistics.

Mieno, T., Neubig, G., Sakti, S., Toda, T., & Nakamura, S. (2015). Speed or accuracy? A study in evaluation of simultaneous speech translation. In *Sixteenth Annual Conference of the International Speech Communication Association.*

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR, abs/1301.3781.*

Mikolov, T., Le, Q. V., & Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Mino, H., Utiyama, M., Sumita, E., & Tokunaga, T. (2017). Key-value attention mechanism for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 290–295, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Mnih, A., & Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 26*, pp. 2265–2273. Curran Associates, Inc.

Mnih, A., & Teh, Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 419–426, USA. Omnipress.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pp. 2204–2212.

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15.

Morishita, M., Oda, Y., Neubig, G., Yoshino, K., Sudoh, K., & Nakamura, S. (2017). An empirical study of mini-batch creation strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 61–68, Vancouver. Association for Computational Linguistics.

Mou, L., Men, R., Li, G., Xu, Y., Zhang, L., Yan, R., & Jin, Z. (2016). Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 130–136. Association for Computational Linguistics.

Müller, M., Rios, A., Voita, E., & Sennrich, R. (2018). A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 61–72, Belgium, Brussels. Association for Computational Linguistics.

Murray, K., & Chiang, D. (2018). Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 212–223, Belgium, Brussels. Association for Computational Linguistics.

Nadejde, M., Reddy, S., Sennrich, R., Dwojak, T., Junczys-Dowmunt, M., Koehn, P., & Birch, A. (2017). Predicting target language CCG supertags improves neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pp. 68–79, Copenhagen, Denmark. Association for Computational Linguistics.

Neco, R. P., & Forcada, M. L. (1997). Asynchronous translations with recurrent neural nets. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, Vol. 4, pp. 2535–2540 vol.4.

Neishi, M., Sakuma, J., Tohda, S., Ishiwatari, S., Yoshinaga, N., & Toyoda, M. (2017). A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size. In *Proceedings of the 4th Workshop on Asian Translation*

(WAT2017), pp. 99–109, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Neubig, G. (2016). Lexicons and minimum risk training for neural machine translation: NAIST-CMU at WAT2016. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pp. 119–125, Osaka, Japan. The COLING 2016 Organizing Committee.

Neubig, G. (2017). Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619*.

Neubig, G., Morishita, M., & Nakamura, S. (2015). Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pp. 35–41, Kyoto, Japan. Workshop on Asian Translation.

Neubig, G., Sperber, M., Wang, X., Felix, M., Matthews, A., Padmanabhan, S., Qi, Y., Sachan, D., Arthur, P., Godard, P., Hewitt, J., Riad, R., & Wang, L. (2018). XNMT: The eXtensible neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pp. 185–192, Boston, MA. Association for Machine Translation in the Americas.

Nguyen, T. Q., & Chiang, D. (2018). Improving lexical choice in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 334–343, New Orleans, Louisiana. Association for Computational Linguistics.

Niehues, J., Cho, E., Ha, T.-L., & Waibel, A. (2016). Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Niehues, J., Cho, E., Ha, T.-L., & Waibel, A. (2017). Analyzing neural MT search and model performance. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 11–17, Vancouver. Association for Computational Linguistics.

Niu, X., Denkowski, M., & Carpuat, M. (2018). Bi-Directional neural machine translation with synthetic parallel data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 84–91, Melbourne, Australia. Association for Computational Linguistics.

Ojha, A. K., Chowdhury, K. D., Liu, C.-H., & Saxena, K. (2018). The RGNLP machine translation systems for WAT 2018. *arXiv preprint arXiv:1812.00798*.

Östling, R., & Tiedemann, J. (2017). Neural machine translation for low-resource languages. *arXiv preprint arXiv:1708.05729*.

Ott, M., Auli, M., Grangier, D., & Ranzato, M. (2018). Analyzing uncertainty in neural machine translation. In Dy, J., & Krause, A. (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 3956–3965, Stockholmsmässan, Stockholm Sweden. PMLR.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*.

Palm, R., Paquet, U., & Winther, O. (2018). Recurrent relational networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 31*, pp. 3368–3378. Curran Associates, Inc.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Parikh, A., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2249–2255, Austin, Texas. Association for Computational Linguistics.

Park, Y., Na, H., Lee, H., Lee, J., & Song, I. (2016). An effective diverse decoding scheme for robust synonymous sentence translation. *AMTA 2016, Vol.*, 53.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In Dasgupta, S., & McAllester, D. (Eds.), *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28 of *Proceedings of Machine Learning Research*, pp. 1310–1318, Atlanta, Georgia, USA. PMLR.

Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M., Ammar, W., Bhagavatula, C., & Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1756–1765. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237. Association for Computational Linguistics.

Pinnis, M., Krišlauks, R., Deksne, D., & Miks, T. (2017). Neural machine translation for morphologically rich languages with improved sub-word units and synthetic data. In Ekštein, K., & Matoušek, V. (Eds.), *Text, Speech, and Dialogue*, pp. 237–245, Cham. Springer International Publishing.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*(1), 77 – 105.

Poncelas, A., Shterionov, D., Way, A., Wenniger, G. M. d. B., & Passban, P. (2018). Investigating backtranslation in neural machine translation. *arXiv preprint arXiv:1804.06189*.

Popel, M., & Bojar, O. (2018). Training tips for the Transformer model. *The Prague Bulletin of Mathematical Linguistics*, *110*(1), 43–70.

Popescu-Belis, A. (2019). Context in neural machine translation: A review of models and evaluations. *arXiv preprint arXiv:1901.09115*.

Pouget-Abadie, J., Bahdanau, D., van Merrienboer, B., Cho, K., & Bengio, Y. (2014). Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 78–85, Doha, Qatar. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding with unsupervised learning. Tech. rep., Technical report, OpenAI.

Ramachandran, P., Liu, P. J., & Le, Q. V. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Ren, S., Zhang, Z., Liu, S., Zhou, M., & Ma, S. (2019). Unsupervised neural machine translation with SMT as posterior regularization. *arXiv preprint arXiv:1901.04112*.

Resnik, P. (1999). Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 527–534, College Park, Maryland, USA. Association for Computational Linguistics.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, San Diego, California. Association for Computational Linguistics.

Rikters, M. (2018). Debugging neural machine translations. *arXiv preprint arXiv:1808.02733*.

Rikters, M., & Fishel, M. (2017). Confidence through attention. *arXiv preprint arXiv:1710.03743*.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1), 1–39.

Rossenbach, N., Rosendahl, J., Kim, Y., Graça, M., Gokrani, A., & Ney, H. (2018). The RWTH Aachen University filtering system for the WMT 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 946–954, Belgium, Brussels. Association for Computational Linguistics.

Ruiz, N., Gangi, M. A. D., Bertoldi, N., & Federico, M. (2017). Assessing the tolerance of neural machine translation systems against speech recognition errors. In *Proc. Interspeech 2017*, pp. 2635–2639.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). *Neurocomputing: Foundations of Research*, chap. Learning Representations by Back-propagating Errors, pp. 696–699. MIT Press, Cambridge, MA, USA.

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Salesky, E., Runge, A., Coda, A., Niehues, J., & Neubig, G. (2018). Optimizing segmentation granularity for neural machine translation. *arXiv preprint arXiv:1810.08641*.

Sankaran, B., Freitag, M., & Al-Onaizan, Y. (2017). Attention-based vocabulary selection for NMT decoding. *arXiv preprint arXiv:1706.03824*.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., & Lillicrap, T. (2017). A simple neural network module for relational reasoning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 4967–4976. Curran Associates, Inc.

Saunders, D., de Gispert, A., Stahlberg, F., & Byrne, B. (2019). Domain adaptive inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.

Saunders, D., Stahlberg, F., de Gispert, A., & Byrne, B. (2018). Multi-representation ensembles and delayed SGD updates improve syntax-based NMT. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 319–325, Melbourne, Australia. Association for Computational Linguistics.

Schmidt, T., & Marg, L. (2018). How to move to neural machine translation for enterprise-scale programs—an early adoption case study..

Schnober, C., Eger, S., Do Dinh, E.-L., & Gurevych, I. (2016). Still not there? Comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1703–1714, Osaka, Japan. The COLING 2016 Organizing Committee.

Schuster, M., & Nakajima, K. (2012). Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Schwarzenberg, R., Harbecke, D., Macketanz, V., Avramidis, E., & Möller, S. (2019). Train, sort, explain: Learning to diagnose translation models. *arXiv preprint arXiv:1903.12017*.

Schwenk, H. (2008). Investigations on large-scale lightly-supervised training for statistical machine translation.. In *International Workshop on Spoken Language Translation (IWSLT) 2008*, pp. 182–189.

Schwenk, H. (2012). Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING 2012: Posters*, pp. 1071–1080, Mumbai, India. The COLING 2012 Organizing Committee.

Schwenk, H., Dechelotte, D., & Gauvain, J.-L. (2006). Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pp. 723–730, Sydney, Australia. Association for Computational Linguistics.

See, A., Luong, M.-T., & Manning, C. D. (2016). Compression of neural machine translation models via pruning. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 291–301, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R. (2017). How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 376–382, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., Birch, A., Currey, A., Germann, U., Haddow, B., Heafield, K., Miceli Barone, A. V., & Williams, P. (2017a). The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pp. 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., & Nadejde, M. (2017b). Nematus: A toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 65–68, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., & Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pp. 83–91, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., & Birch, A. (2016a). Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pp. 371–376, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., & Birch, A. (2016b). 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)*, pp. 86–96, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., & Birch, A. (2016c). 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)*, pp. 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shah, H., & Barber, D. (2018). Generative neural machine translation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 31*, pp. 1346–1355. Curran Associates, Inc.

Shang, L., Lu, Z., & Li, H. (2015). Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1577–1586, Beijing, China. Association for Computational Linguistics.

Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., & Liu, Y. (2016). Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., & Zhang, C. (2018a). DiSAN: Directional self-attention network for RNN/CNN-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shen, T., Zhou, T., Long, G., Jiang, J., Wang, S., & Zhang, C. (2018b). Reinforced self-attention network: A hybrid of hard and soft attention for sequence modeling. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 4345–4352. AAAI Press.

Skorokhodov, I., Rykachevskiy, A., Emelyanenko, D., Slotin, S., & Ponkratov, A. (2018). Semi-supervised neural machine translation with language models. In *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)*, pp. 37–44, Boston, MA. Association for Machine Translation in the Americas.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive Autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 151–161, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Sønderby, S. K., Sønderby, C. K., Nielsen, H., & Winther, O. (2015). Convolutional LSTM networks for subcellular localization of proteins. In Dediu, A.-H., Hernández-Quiroz, F., Martín-Vide, C., & Rosenblueth, D. A. (Eds.), *Algorithms for Computational Biology*, pp. 68–80, Cham. Springer International Publishing.

Sountsov, P., & Sarawagi, S. (2016). Length bias in encoder decoder models and a case for global conditioning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1516–1525, Austin, Texas. Association for Computational Linguistics.

Specia, L. (2011). Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pp. 73–80.

Specia, L., Blain, F., Logacheva, V., Astudillo, R., & Martins, A. F. T. (2018). Findings of the WMT 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 689–709, Belgium, Brussels. Association for Computational Linguistics.

Sperber, M., Neubig, G., Niehues, J., & Waibel, A. (2017). Neural lattice-to-sequence models for uncertain inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1380–1389, Copenhagen, Denmark. Association for Computational Linguistics.

Sproat, R., & Jaitly, N. (2016). RNN approaches to text normalization: A challenge. *arXiv preprint arXiv:1611.00068*.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Stahlberg, F., & Byrne, B. (2017). Unfolding and shrinking neural machine translation ensembles. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1946–1956, Copenhagen, Denmark. Association for Computational Linguistics.

Stahlberg, F., & Byrne, B. (2019). On NMT search errors and model errors: Cat got your tongue?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong. Association for Computational Linguistics.

Stahlberg, F., Cross, J., & Stoyanov, V. (2018a). Simple fusion: Return of the language model. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 204–211, Belgium, Brussels. Association for Computational Linguistics.

Stahlberg, F., de Gispert, A., & Byrne, B. (2018b). The University of Cambridge's machine translation systems for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 504–512, Belgium, Brussels. Association for Computational Linguistics.

Stahlberg, F., de Gispert, A., Hasler, E., & Byrne, B. (2017). Neural machine translation by minimising the Bayes-risk with respect to syntactic translation lattices. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 362–368, Valencia, Spain. Association for Computational Linguistics.

Stahlberg, F., Hasler, E., & Byrne, B. (2016). The edit distance transducer in action: The University of Cambridge English-German system at WMT16. In *Proceedings of the*

First Conference on Machine Translation, pp. 377–384, Berlin, Germany. Association for Computational Linguistics.

Stahlberg, F., Hasler, E., Saunders, D., & Byrne, B. (2017). SGNMT – a flexible NMT decoding platform for quick prototyping of new models and search strategies. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 25–30, Copenhagen, Denmark. Association for Computational Linguistics.

Stahlberg, F., Hasler, E., Waite, A., & Byrne, B. (2016). Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 299–305, Berlin, Germany. Association for Computational Linguistics.

Stahlberg, F., Saunders, D., & Byrne, B. (2018). An operation sequence model for explainable neural machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 175–186, Brussels, Belgium. Association for Computational Linguistics.

Stahlberg, F., Saunders, D., de Gispert, A., & Byrne, B. (2019). CUED@WMT19:EWC&LMs. In *Proceedings of the Fourth Conference on Machine Translation: Shared Task Papers*. Association for Computational Linguistics.

Stahlberg, F., Saunders, D., Iglesias, G., & Byrne, B. (2018). Why not be versatile? Applications of the SGNMT decoder for machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pp. 208–216, Boston, MA. Association for Machine Translation in the Americas.

Stern, M., Chan, W., Kiros, J. R., & Uszkoreit, J. (2019). Insertion Transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*.

Stern, M., Shazeer, N., & Uszkoreit, J. (2018). Blockwise parallel decoding for deep autoregressive models. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 31*, pp. 10086–10095. Curran Associates, Inc.

Su, J., Tan, Z., Xiong, D., Ji, R., Shi, X., & Liu, Y. (2017). Lattice-based recurrent neural network encoders for neural machine translation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 3302–3308. AAAI Press.

Su, J., Wu, S., Xiong, D., Lu, Y., Han, X., & Zhang, B. (2018). Variational recurrent neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 28*, pp. 2440–2448. Curran Associates, Inc.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. Curran Associates, Inc.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Takase, S., & 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)*, pp. 3999–4004, Minneapolis, Minnesota. Association for Computational Linguistics.

Tamchyna, A., Weller-Di Marco, M., & Fraser, A. (2017). Modeling target-side inflection in neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pp. 32–42, Copenhagen, Denmark. Association for Computational Linguistics.

Tang, G., Müller, M., Rios, A., & Sennrich, R. (2018a). Why self-attention? A targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4263–4272, Brussels, Belgium. Association for Computational Linguistics.

Tang, G., Sennrich, R., & Nivre, J. (2018b). An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 26–35, Belgium, Brussels. Association for Computational Linguistics.

Tang, Y., Meng, F., Lu, Z., Li, H., & Yu, P. L. (2016). Neural machine translation with external phrase memory. *arXiv preprint arXiv:1606.01792*.

Thompson, B., Gwinnup, J., Khayrallah, H., Duh, K., & Koehn, P. (2019). Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

Tong, A., Diduch, L., Fiscus, J., Haghpanah, Y., Huang, S., Joy, D., Peterson, K., & Soboroff, I. (2018). Overview of the NIST 2016 LoReHLT evaluation. *Machine Translation*, *32*(1-2), 11–30.

Toral, A., & Sánchez-Cartagena, V. M. (2017). A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 1063–1073, Valencia, Spain. Association for Computational Linguistics.

Tromble, R., Kumar, S., Och, F. J., & Macherey, W. (2008). Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 620–629, Honolulu, Hawaii. Association for Computational Linguistics.

Tu, Z., Liu, Y., Shang, L., Liu, X., & Li, H. (2017). Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 3097–3103. AAAI Press.

Tu, Z., Lu, Z., Liu, Y., Liu, X., & 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)*, pp. 76–85, Berlin, Germany. Association for Computational Linguistics.

Ueffing, N., & Ney, H. (2005). Word-level confidence estimation for machine translation using phrase-based translation models. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 763–770, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Upadhyay, S., Faruqui, M., Dyer, C., & Roth, D. (2016). Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1661–1670, Berlin, Germany. Association for Computational Linguistics.

van den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1747–1756. JMLR.org.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., & Uszkoreit, J. (2018). Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pp. 193–199, Boston, MA. Association for Machine Translation in the Americas.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D., & Batra, D. (2016). Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Vogel, S., Ney, H., & Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Volkart, L., Bouillon, P., & Girletti, S. (2018). Statistical vs. neural machine translation: A comparison of mth and deepl at swiss post's language service. In *Proceedings of the 40th Conference Translating and the Computer*, pp. 145–150, London, United-Kingdom.

Waibel, A., Hanazawa, T., Hinton, G. E., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(3), 328–339.

Wang, C., Li, M., & Smola, A. (2019). Language models with Transformers. *arXiv preprint arXiv:1904.09408*.

Wang, C., Zhang, J., & Chen, H. (2018a). Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 479–488, Brussels, Belgium. Association for Computational Linguistics.

Wang, M., Gong, L., Zhu, W., Xie, J., & Bian, C. (2018b). Tencent neural machine translation systems for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 522–527, Belgium, Brussels. Association for Computational Linguistics.

Wang, M., Lu, Z., Li, H., & Liu, Q. (2016). Memory-enhanced decoder for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 278–286, Austin, Texas. Association for Computational Linguistics.

Wang, X., Utiyama, M., & Sumita, E. (2018). CytonMT: An efficient neural machine translation open-source toolkit implemented in C++. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 133–138, Brussels, Belgium. Association for Computational Linguistics.

Wang, X., Lu, Z., Tu, Z., Li, H., Xiong, D., & Zhang, M. (2017). Neural machine translation advised by statistical machine translation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 3330–3336. AAAI Press.

Wang, X., Tu, Z., & Zhang, M. (2018a). Incorporating statistical machine translation word knowledge into neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(12), 2255–2266.

Wang, X., Pham, H., Dai, Z., & Neubig, G. (2018b). SwitchOut: An efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 856–861, Brussels, Belgium. Association for Computational Linguistics.

Wang, Y., Xia, Y., Zhao, L., Bian, J., Qin, T., Liu, G., & Liu, T.-Y. (2018c). Dual transfer learning for neural machine translation with marginal distribution regularization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, Y., Tian, F., He, D., Qin, T., Zhai, C., & Liu, T.-Y. (2019). Non-autoregressive machine translation with auxiliary regularization. *arXiv preprint arXiv:1902.10245*.

Wang, Y., Cheng, S., Jiang, L., Yang, J., Chen, W., Li, M., Shi, L., Wang, Y., & Yang, H. (2017). Sogou neural machine translation systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pp. 410–415, Copenhagen, Denmark. Association for Computational Linguistics.

Welleck, S., Brantley, K., Daumé III, H., & Cho, K. (2019). Non-monotonic sequential text generation. *arXiv preprint arXiv:1902.02192*.

Werlen, L. M., Pappas, N., Ram, D., & Popescu-Belis, A. (2018). Global-context neural machine translation through target-side attentive residual connections. *researchgate.net*.

Wieting, J., & Kiela, D. (2019). No training required: Exploring random encoders for sentence classification. *arXiv preprint arXiv:1901.10444*.

Wiseman, S., & Rush, A. M. (2016). Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1296–1306, Austin, Texas. Association for Computational Linguistics.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, *23*(3), 377–403.

Wu, F., Fan, A., Baevski, A., Dauphin, Y. N., & Auli, M. (2019). Pay less attention with lightweight and dynamic convolutions. In *ICLR*.

Wu, L., Tian, F., Qin, T., Lai, J., & Liu, T.-Y. (2018). A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3612–3621, Brussels, Belgium. Association for Computational Linguistics.

Wu, L., Xia, Y., Zhao, L., Tian, F., Qin, T., Lai, J., & Liu, T.-Y. (2017). Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*.

Wu, W., Wang, H., Liu, T., & Ma, S. (2018). Phrase-level self-attention networks for universal sentence encoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3729–3738, Brussels, Belgium. Association for Computational Linguistics.

Wu, Y., & Zhao, H. (2018). Finding better subword segmentation for neural machine translation. In Sun, M., Liu, T., Wang, X., Liu, Z., & Liu, Y. (Eds.), *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pp. 53–64, Cham. Springer International Publishing.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiong, H., He, Z., Hu, X., & Wu, H. (2018). Multi-channel encoder for neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xu, H., & Liu, Q. (2019). Neutron: An implementation of the Transformer translation model and its variants. *arXiv preprint arXiv:1903.07402*.

Xu, K., Ba, J. L., Kiros, J. R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057.

Yang, J., Zhang, B., Qin, Y., Zhang, X., Lin, Q., & Su, J. (2018a). Otem&utem: Over- and under-translation evaluation metric for nmt. In Zhang, M., Ng, V., Zhao, D., Li, S., & Zan, H. (Eds.), *Natural Language Processing and Chinese Computing*, pp. 291–302, Cham. Springer International Publishing.

Yang, Y., Huang, L., & Ma, M. (2018b). Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3054–3059, Brussels, Belgium. Association for Computational Linguistics.

Yang, Z., Chen, L., & Le Nguyen, M. (2018c). Regularizing forward and backward decoding to improve neural machine translation. In *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 73–78.

Yang, Z., Chen, W., Wang, F., & Xu, B. (2016). A character-aware encoder for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3063–3070, Osaka, Japan. The COLING 2016 Organizing Committee.

Yang, Z., Chen, W., Wang, F., & Xu, B. (2018). Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1346–1355, New Orleans, Louisiana. Association for Computational Linguistics.

Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2016). Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 21–29.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., & Courville, A. (2015). Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pp. 4507–4515.

Yu, L., Blunsom, P., Dyer, C., Grefenstette, E., & Kocisky, T. (2016). The neural noisy channel. *arXiv preprint arXiv:1611.02554*.

Yu, L., d'Autume, C. d. M., Dyer, C., Blunsom, P., Kong, L., & Ling, W. (2018). Sentence encoding with tree-constrained relation networks. *arXiv preprint arXiv:1811.10475*.

Yu, L., Sartran, L., Stokowiec, W., Ling, W., Kong, L., Blunsom, P., & Dyer, C. (2020). Better document-level machine translation with bayes' rule. *Transactions of the Association for Computational Linguistics*, *8*, 346–360.

Yuan, Z., & Briscoe, T. (2016). Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 380–386, San Diego, California. Association for Computational Linguistics.

Zamora-Martinez, F., Castro-Bleda, M. J., & Schwenk, H. (2010). N-gram-based machine translation enhanced with neural networks for the French-English BTEC-IWSLT'10 task. In *International Workshop on Spoken Language Translation (IWSLT) 2010*.

Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zenkel, T., Wuebker, J., & DeNero, J. (2019). Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

Zhang, B., Xiong, D., Su, J., Duan, H., & Zhang, M. (2016). Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 521–530, Austin, Texas. Association for Computational Linguistics.

Zhang, D., Crego, J., & Senellart, J. (2018). Analyzing knowledge distillation in neural machine translation. In *International Workshop on Spoken Language Translation IWSLT*.

Zhang, J., Ding, Y., Shen, S., Cheng, Y., Sun, M., Luan, H., & Liu, Y. (2017). THUMT: An open source toolkit for neural machine translation. *arXiv preprint arXiv:1706.06415*.

Zhang, J., & Zong, C. (2016a). Bridging neural machine translation and bilingual dictionaries. *arXiv preprint arXiv:1610.07272*.

Zhang, J., & Zong, C. (2016b). Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1535–1545, Austin, Texas. Association for Computational Linguistics.

Zhang, J., Utiyama, M., Sumita, E., Neubig, G., & Nakamura, S. (2017). Improving neural machine translation through phrase-based forced decoding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 152–162, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zhang, Q., Liang, S., & Yilmaz, E. (2018a). Variational self-attention model for sentence representation. *arXiv preprint arXiv:1812.11559*.

Zhang, Z., Liu, S., Li, M., Zhou, M., & Chen, E. (2018b). Bidirectional generative adversarial networks for neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 190–199, Brussels, Belgium. Association for Computational Linguistics.

Zhang, Z., Liu, S., Li, M., Zhou, M., & Chen, E. (2018c). Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhang, Z., Wu, S., Liu, S., Li, M., Zhou, M., & Chen, E. (2018d). Regularizing neural machine translation by target-bidirectional agreement. *arXiv preprint arXiv:1808.04064*.

Zhang, Z., Wang, R., Utiyama, M., Sumita, E., & Zhao, H. (2018e). Exploring recombination for efficient decoding of neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4785–4790, Brussels, Belgium. Association for Computational Linguistics.

Zhou, L., Hu, W., Zhang, J., & Zong, C. (2017). Neural system combination for machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 378–384, Vancouver, Canada. Association for Computational Linguistics.

Zipf, G. K. (1946). The psychology of language. In *Encyclopedia of psychology*, pp. 332–341. Philosophical Library.