An Operation Sequence Model for Explainable Neural Machine Translation

Felix Stahlberg and **Danielle Saunders** and **Bill Byrne**
Department of Engineering
University of Cambridge, UK
{fs439,ds636,wjb31}@cam.ac.uk

Abstract

We propose to achieve explainable neural machine translation (NMT) by changing the output representation to explain itself. We present a novel approach to NMT which generates the target sentence by monotonically walking through the source sentence. Word reordering is modeled by operations which allow setting markers in the target sentence and move a target-side write head between those markers. In contrast to many modern neural models, our system emits explicit word alignment information which is often crucial to practical machine translation as it improves explainability. Our technique can outperform a plain text system in terms of BLEU score under the recent Transformer architecture on Japanese-English and Portuguese-English, and is within 0.5 BLEU difference on Spanish-English.

1 Introduction

Neural machine translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) are remarkably effective in modelling the distribution over target sentences conditioned on the source sentence, and yield superior translation performance compared to traditional statistical machine translation (SMT) on many language pairs. However, it is often difficult to extract a comprehensible explanation for the predictions of these models as information in the network is represented by real-valued vectors or matrices (Ding et al., 2017). In contrast, the translation process in SMT is 'transparent' as it can identify the source word which caused a target word through word alignment. Most NMT models do not use the concept of word alignment. It is tempting to interpret encoder-decoder attention matrices (Bahdanau et al., 2015) in neural models as (soft) alignments, but previous work has found that the attention weights in NMT are often erratic (Cheng et al., 2016) and differ significantly from traditional word alignments (Koehn and Knowles, 2017; Ghader and Monz, 2017). We will discuss the difference between attention and alignment in detail in Sec. 4. The goal of this paper is explainable NMT by developing a transparent translation process for neural models. Our approach does not change the neural architecture, but represents the translation together with its alignment as a linear sequence of operations. The neural model predicts this operation sequence, and thus simultaneously generates a translation and an explanation for it in terms of alignments from the target words to the source words that generate them. The operation sequence is "self-explanatory"; it does not explain an underlying NMT system but is rather a single representation produced by the NMT system that can be used to generate translations along with an accompanying explanatory alignment to the source sentence. We report competitive results of our method on Spanish-English, Portuguese-English, and Japanese-English, with the benefit of producing hard alignments for better interpretability. We discuss the theoretical connection between our approach and hierarchical SMT (Chiang, 2005) by showing that an operation sequence can be seen as a derivation in a formal grammar.

2 A Neural Operation Sequence Model

Our operation sequence neural machine translation (OSNMT) model is inspired by the operation sequence model for SMT (Durrani et al., 2011), but changes the set of operations to be more appropriate for neural sequence models. OSNMT is not restricted to a particular architecture, i.e. any seq2seq model such as RNN-based, convolutional, or self-attention-based models (Bahdanau et al.,

2015; Vaswani et al., 2017; Gehring et al., 2017) could be used. In this paper, we use the recent Transformer model architecture (Vaswani et al., 2017) in all experiments.

In OSNMT, the neural seq2seq model learns to produce a sequence of operations. An OSNMT operation sequence describes a translation (the 'compiled' target sentence) and explains each target token with a hard link into the source sentence. OSNMT keeps track of the positions of a source-side read head and a target-side write head. The read head monotonically walks through the source sentence, whereas the position of the write head can be moved from marker to marker in the target sentence. OSNMT defines the following operations to control head positions and produce output words.

- `POP_SRC`: Move the read head right by one token.
- `SET_MARKER`: Insert a marker symbol into the target sentence at the position of the write head.
- `JMP_FWD`: Move the write head to the nearest marker right of the current head position in the target sentence.
- `JMP_BWD`: Move the write head to the nearest marker left of the current head position in the target sentence.
- `INSERT`(t): Insert a target token t into the target sentence at the position of the write head.

Tab. 1 illustrates the generation of a Japanese-English translation in detail. The neural seq2seq model is trained to produce the sequence of operations in the first column of Tab. 1. The initial state of the target sentence is a single marker symbol X_1. Generative operations like `SET_MARKER` or `INSERT`(t) insert a single symbol left of the current marker (highlighted). The model begins with a `SET_MARKER` operation, which indicates that the translation of the first word in the source sentence is not at the beginning of the target sentence. Indeed, after "translating" the identities '2000' and 'hr', in time step 6 the model jumps back to the marker X_2 and continues writing left of '2000'. The translation process terminates when the read head is at the end of the source sentence. The final translation in plain text can be obtained by removing all markers from the (compiled) target sentence.

2.1 OSNMT Represents Alignments

The word alignment can be derived from the operation sequence by looking up the position of the read head for each generated target token. The alignment for the example in Tab. 1 is shown in Fig. 1. Note that similarly to the IBM models (Brown et al., 1993) and the OSM for SMT (Durrani et al., 2011), our OSNMT can only represent 1:n alignments. Thus, each target token is aligned to exactly one source token, but a source token can generate any number of (possibly non-consecutive) target tokens.

2.2 OSNMT Represents Hierarchical Structure

We can also derive a tree structure from the operation sequence in Tab. 1 (Fig. 2) in which each marker is represented by a nonterminal node with outgoing arcs to symbols inserted at that marker. The target sentence can be read off the tree by depth-first search traversal (post-order).

More formally, synchronous context-free grammars (SCFGs) generate pairs of strings by pairing two context-free grammars. Phrase-based hierarchical SMT (Chiang, 2005) uses SCFGs to model the relation between the source sentence and the target sentence. Multitext grammars (MTGs) are a generalization of SCFGs to more than two output streams (Melamed, 2003; Melamed et al., 2004). We find that an OSNMT sequence can be interpreted as sequence of rules of a tertiary MTG $\mathcal{G}$ which generates 1.) the source sentence, 2.) the target sentence, and 3.) the position of the target side write head. The start symbol of $\mathcal{G}$ is

$$[(S), (X_1), (P_1)]^{\mathsf{T}} \tag{1}$$

which initializes the source sentence stream with a single nonterminal S, the target sentence with the initial marker X_1 and the position of the write head with 1 (P_1). Following Melamed et al. (2004) we denote rules in $\mathcal{G}$ as

$$[(\alpha_1), (\alpha_2), (\alpha_3)]^{\mathsf{T}} \rightarrow [(\beta_1), (\beta_2), (\beta_3)]^{\mathsf{T}} \tag{2}$$

where $\alpha_1, \alpha_2, \alpha_3$ are single nonterminals or empty, $\beta_1, \beta_2, \beta_3$ are strings of terminals and nonterminals, and $\alpha_i \rightarrow \beta_i$ for all $i \in \{1, 2, 3\}$ with nonempty α_i are the rewriting rules for each of

	Operation	Source sentence	Target sentence (compiled)
		2000 hr の 安定 動作 を 確認 し た	X_1
1	SET_MARKER	2000 hr の 安定 動作 を 確認 し た	X_2 X_1
2	2000	2000 hr の 安定 動作 を 確認 し た	X_2 2000 X_1
3	POP_SRC	2000 hr の 安定 動作 を 確認 し た	X_2 2000 X_1
4	hr	2000 hr の 安定 動作 を 確認 し た	X_2 2000 hr X_1
5	POP_SRC	2000 hr の 安定 動作 を 確認 し た	X_2 2000 hr X_1
6	JMP_BWD	2000 hr の 安定 動作 を 確認 し た	X_2 2000 hr X_1
7	SET_MARKER	2000 hr の 安定 動作 を 確認 し た	X_3 X_2 2000 hr X_1
8	of	2000 hr の 安定 動作 を 確認 し た	X_3 of X_2 2000 hr X_1
9	POP_SRC	2000 hr の 安定 動作 を 確認 し た	X_3 of X_2 2000 hr X_1
10	JMP_BWD	2000 hr の 安定 動作 を 確認 し た	X_3 of X_2 2000 hr X_1
11	stable	2000 hr の 安定 動作 を 確認 し た	stable X_3 of X_2 2000 hr X_1
12	POP_SRC	2000 hr の 安定 動作 を 確認 し た	stable X_3 of X_2 2000 hr X_1
13	operation	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr X_1
14	POP_SRC	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr X_1
15	JMP_FWD	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr X_1
16	JMP_FWD	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr X_1
17	was	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr was X_1
18	POP_SRC	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr was X_1
19	POP_SRC	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr was X_1
20	confirmed	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr was confirmed X_1
21	POP_SRC	2000 hr の 安定 動作 を 確認 し た	stable operation X_3 of X_2 2000 hr was confirmed X_1

Table 1: Generation of the target sentence "stable operation of 2000 hr was confirmed" from the source sentence "2000 hr の 安定 動作 を 確認 し た". The neural model produces the linear sequence of operations in the first column. The positions of the source-side read head and the target-side write head are highlighted. The marker in the target sentence produced by the i-th SET_MARKER operation is denoted with 'X_{i+1}'; X_1 is the initial marker. We denote INSERT(t) operations as t to simplify notation.

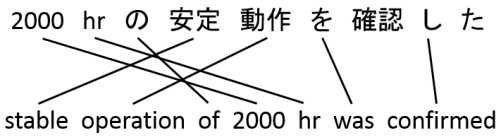


Figure 1: The translation and the alignment derived from the operation sequence in Tab. 1.

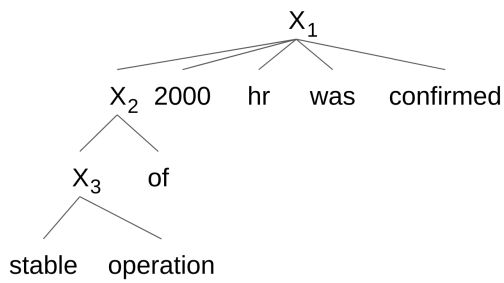


Figure 2: Target-side tree representation of the operation sequence in Tab. 1.

the three individual components which need to be applied simultaneously. POP_SRC extends the source sentence prefix in the first stream by one token.

$$\texttt{POP\_SRC} : \forall s \in \mathcal{V}_{src} : \begin{bmatrix} (S) \\ () \\ () \end{bmatrix} \rightarrow \begin{bmatrix} (sS) \\ () \\ () \end{bmatrix} \quad (3)$$

where $\mathcal{V}_{src}$ is the source language vocabulary. A jump from marker X_i to X_j is realized by replacing P_i with P_j in the third grammar component:

$$\texttt{JMP} : \forall i, j \in \mathcal{N} : [(), (), P_i]^{\mathsf{T}} \rightarrow [(), (), (iP_j)]^{\mathsf{T}} \quad (4)$$

where $\mathcal{N} = \{k \in \mathbb{N} | k \leq n\}$ is the set of the first n natural numbers for a sufficiently large n. The generative operations (SET_MARKER and

Derivation	OSNMT
$[(S), (X_1), P_1]^\mathsf{T}$	
$\overset{\text{Eq. 3}}{\rightarrow} [(2000\ S), (X_1), P_1]^\mathsf{T}$	SET_MARKER
$\overset{\text{Eq. 5}}{\rightarrow} [(2000\ S), (X_2 X_1), (P_1)]^\mathsf{T}$	2000
$\overset{\text{Eq. 6}}{\rightarrow} [(2000\ S), (X_2\ 2000\ X_1), (P_1)]^\mathsf{T}$	POP_SRC
$\overset{\text{Eq. 3}}{\rightarrow} \begin{bmatrix} (2000\ \text{hr}\ S) \\ (X_2\ 2000\ X_1) \\ (P_1) \end{bmatrix}$	hr
$\overset{\text{Eq. 6}}{\rightarrow} \begin{bmatrix} (2000\ \text{hr}\ S) \\ (X_2\ 2000\ \text{hr}\ X_1) \\ (P_1) \end{bmatrix}$	POP_SRC
$\overset{\text{Eq. 3}}{\rightarrow} \begin{bmatrix} (2000\ \text{hr}\ \curvearrowleft\ S) \\ (X_2\ 2000\ \text{hr}\ X_1) \\ (P_1) \end{bmatrix}$	JMP_BWD
$\overset{\text{Eq. 4}}{\rightarrow} \begin{bmatrix} (2000\ \text{hr}\ \curvearrowleft\ S) \\ (X_2\ 2000\ \text{hr}\ X_1) \\ (1\ P_2) \end{bmatrix}$	SET_MARKER
$\overset{\text{Eq. 5}}{\rightarrow} \begin{bmatrix} (2000\ \text{hr}\ \curvearrowleft\ S) \\ (X_3 X_2\ 2000\ \text{hr}\ X_1) \\ (1\ P_2) \end{bmatrix}$	of
$\overset{\text{Eq. 6}}{\rightarrow} \begin{bmatrix} (2000\ \text{hr}\ \curvearrowleft\ S) \\ (X_3\ \text{of}\ X_2\ 2000\ \text{hr}\ X_1) \\ (1\ P_2) \end{bmatrix}$	...
...	

Table 2: Derivation in $\mathcal{G}$ for the example of Tab. 1.

INSERT(t)) insert symbols into the second component.

$$\texttt{SET\_MARKER} : \forall i \in \mathcal{N} : \begin{bmatrix} () \\ (X_i) \\ (P_i) \end{bmatrix} \rightarrow \begin{bmatrix} () \\ (X_{i+1} X_i) \\ (P_i) \end{bmatrix} \tag{5}$$

$$\texttt{INSERT} : \forall i \in \mathcal{N}, t \in \mathcal{V}_{trg} : \begin{bmatrix} () \\ (X_i) \\ (P_i) \end{bmatrix} \rightarrow \begin{bmatrix} () \\ (t X_i) \\ (P_i) \end{bmatrix} \tag{6}$$

where $\mathcal{V}_{trg}$ is the target language vocabulary. The identity mapping $P_i \rightarrow P_i$ in the third component enforces that the write head is at marker X_i. We note that $\mathcal{G}$ is not only context-free but also regular in the first and third components (but not in the second component due to Eq. 5). Rules of the form in Eq. 6 are directly related to alignment links (cf. Fig. 1) as they represent the fact that target token t is aligned to the last terminal symbol in the first stream. We formalize removing markers/nonterminals at the end by introducing a special nonterminal T which is eventually mapped to the end-of-sentence symbol EOS:

$$[(S), (), ()]^\mathsf{T} \rightarrow [(T), (), ()]^\mathsf{T} \tag{7}$$

$$[(T), (), ()]^\mathsf{T} \rightarrow [(\text{EOS}), (), ()]^\mathsf{T} \tag{8}$$

$$\forall i \in \mathcal{N} : [(T), (X_i), ()]^\mathsf{T} \rightarrow [(T), (\epsilon), ()]^\mathsf{T} \tag{9}$$

$$\forall i \in \mathcal{N} : [(T), (), (P_i)]^\mathsf{T} \rightarrow [(T), (), (\epsilon)]^\mathsf{T} \tag{10}$$

Tab. 2 illustrates that there is a 1:1 correspondence between a derivation in $\mathcal{G}$ and an OSNMT operation sequence. The target-side derivation (the second component in $\mathcal{G}$) is structurally similar to a binarized version of the tree in Fig. 2. However, we assign scores to the structure via the corresponding OSNMT sequence which does not need to obey the usual conditional independence assumptions in hierarchical SMT. Therefore, even though $\mathcal{G}$ is context-free in the second component, our scoring model for $\mathcal{G}$ is more powerful as it conditions on the OSNMT history which potentially contains context information. Note that OSNMT is deficient (Brown et al., 1993) as it assigns non-zero probability mass to any operation sequence, not only those with derivation in G.

We further note that subword-based OSNMT can potentially represent any alignment to any target sentence as long as the alignment does not violate the 1:n restriction. This is in contrast to phrase-based SMT where reference translations often do not have a derivation in the SMT system due to coverage problems (Auli et al., 2009).

2.3 Comparison to the OSM for SMT

Our OSNMT set of operations (POP_SRC, SET_MARKER, JMP_FWD, JMP_BWD, and INSERT(t)) is inspired by the original OSM for SMT (Durrani et al., 2011) as it also represents the translation process as linear sequence of operations. However, there are significant differences which make OSNMT more suitable for neural models. First, OSNMT is monotone on the source side, and allows jumps on the target side. SMT-OSM operations jump in the source sentence. We argue that source side monotonicity potentially mitigates coverage issues of neural models (over- and under-translation (Tu et al., 2016)) as the attention can learn to scan the source sentence from left to right. Another major difference is that we use *markers* rather than *gaps*, and do not close a gap/marker after jumping to it. This is an implication of OSNMT jumps being defined on the target side since the size of a span is unknown at inference time.

Algorithm 1 Align2OSNMT(a, $\mathbf{x}$, $\mathbf{y}$)

$holes \leftarrow \{(0, \infty)\}$
$ops \leftarrow \langle\rangle$ {Initialize with empty list}
$head \leftarrow 0$
for $i \leftarrow 1$ **to** $|\mathbf{x}|$ **do**
  for all $j \in \{j | a_j = i\}$ **do**
    $hole\_idx \leftarrow holes$.find($j$)
    $d \leftarrow hole\_idx - head$
    if $d < 0$ **then**
      ops.extend(JMP_BWD.repeat($-d$))
    end if
    if $d > 0$ **then**
      ops.extend(JMP_FWD.repeat(d))
    end if
    $head \leftarrow hole\_idx$
    $(s, t) \leftarrow holes[head]$
    if $s \neq j$ **then**
      $holes$.append($(s, j-1)$)
      $head \leftarrow head + 1$
      ops.append(SET_MARKER)
    end if
    ops.append($\mathtt{y_j}$)
    $holes[head] \leftarrow (j+1, t)$
  end for
  ops.append(SRC_POP)
end for
return ops

3 Training

We train our Transformer model as usual by minimising the negative log-likelihood of the target sequence. However, in contrast to plain text NMT, the target sequence is not a plain sequence of subword or word tokens but a sequence of operations. Consequently, we need to map the target sentences in the training corpus to OSNMT representations. We first run a statistical word aligner like Giza++ (Och and Ney, 2003) to obtain an aligned training corpus. We delete all alignment links which violate the 1:n restriction of OSNMT (cf. Sec. 2). The alignments together with the target sentences are then used to generate the reference operation sequences for training. The algorithm for this conversion is shown in Alg. 1.[1] Note that an operation sequence represents one specific alignment, which means that the only way for an OSNMT sequence to be generated correctly is if both the word alignment and the target sentence are also correct. Thereby, the neural model learns to align and translate at the same time. However, there is spurious ambiguity as one alignment can be represented by different OSNMT sequences. For instance, simply adding a SET_MARKER operation at the end of an OSNMT sequence does not change the alignment represented by it.

[1] A Python implementation is available at https://github.com/fstahlberg/ucam-scripts/blob/master/t2t/align2osm.py.

Corpus	Language pair	# Sentences
Scielo	Spanish-English	587K
Scielo	Portuguese-English	513K
WAT	Japanese-English	1M

Table 3: Training set sizes.

4 Results

We evaluate on three language pairs: Japanese-English (ja-en), Spanish-English (es-en), and Portuguese-English (pt-en). We use the ASPEC corpus (Nakazawa et al., 2016) for ja-en and the health science portion of the Scielo corpus (Neves and Névéol, 2016) for es-en and pt-en. Training set sizes are summarized in Tab. 3. We use byte pair encoding (Sennrich et al., 2016) with 32K merge operations for all systems (joint encoding models for es-en and pt-en and separate source/target models for ja-en). We trained Transformer models (Vaswani et al., 2017)[2] until convergence (250K steps for plain text, 350K steps for OSNMT) on a single GPU using Tensor2Tensor (Vaswani et al., 2018) after removing sentences with more than 250 tokens. Batches contain around 4K source and 4K target tokens. Transformer training is very sensitive to the batch size and the number of GPUs (Popel and Bojar, 2018). Therefore, we delay SGD updates (Saunders et al., 2018) to every 8 steps to simulate 8 GPU training as recommended by Vaswani et al. (2017). Based on the performance on the ja-en dev set we decode the plain text systems with a beam size of 4 and OSNMT with a beam size of 8 using our SGNMT decoder (Stahlberg et al., 2017). We use length normalization for ja-en but not for es-en or pt-en. We report cased multi-bleu.pl BLEU scores on the tokenized text to be comparable with the WAT evaluation campaign on ja-en.[3].

[2] We follow the transformer_base configuration and use 6 layers, 512 hidden units, and 8 attention heads in both the encoder and decoder.

[3] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=2&o=4

Method	BLEU es-en	BLEU pt-en
Align on subword level	36.7	38.1
Convert word level alignments	37.1	38.4

Table 4: Generating training alignments on the subword level.

Type	Frequency
Valid	**92.49%**
Not enough SRC_POP	7.28%
Too many SRC_POP	0.22%
Write head out of range	0.06%

Table 5: Frequency of invalid OSNMT sequences produced by an unconstrained decoder on the ja-en test set.

Generating training alignments As outlined in Sec. 3 we use Giza++ (Och and Ney, 2003) to generate alignments for training OSNMT. We experimented with two different methods to obtain alignments on the subword level. First, Giza++ can directly align the source-side subword sequences to target-side subword sequences. Alternatively, we can run Giza++ on the word level, and convert the word alignments to subword alignments in a post-processing step by linking subwords if the words they belong to are aligned with each other. Tab. 4 compares both methods and shows that converting word alignments is marginally better. Thus, we use this method in all other experiments.

Constrained beam search Unconstrained neural decoding can yield invalid OSNMT sequences. For example, the JMP_FWD and JMP_BWD operations are undefined if the write head is currently at the position of the last or first marker, respectively. The number of SRC_POP operations must be equal to the number of source tokens in order for the read head to scan the entire source sentence. Therefore, we constrain these operations during decoding. We have implemented the constraints in our publicly available SGNMT decoding platform (Stahlberg et al., 2017). However, these constraints are only needed for a small fraction of the sentences. Tab. 5 shows that even unconstrained decoding yields valid OSNMT sequences in 92.49% of the cases.

Comparison with plain text NMT Tab. 6 compares our OSNMT systems with standard plain text models on all three language pairs. OSNMT performs better on the pt-en and ja-en test sets, but

Representation	BLEU es-en	BLEU pt-en	BLEU ja-en dev	BLEU ja-en test
Plain	37.6	37.5	28.3	28.1
OSNMT	37.1	38.4	28.1	28.8

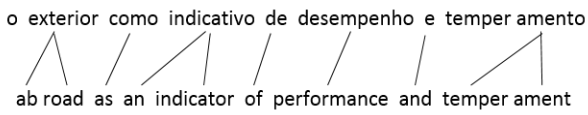
Table 6: Comparison between plain text and OSNMT on Spanish-English (es-en), Portuguese-English (pt-en), and Japanese-English (ja-en).

slightly worse on es-en. We think that more engineering work such as optimizing the set of operations or improving the training alignments could lead to more consistent gains from using OSNMT. However, we leave this to future work since the main motivation for this paper is explainable NMT and not primarily improving translation quality.

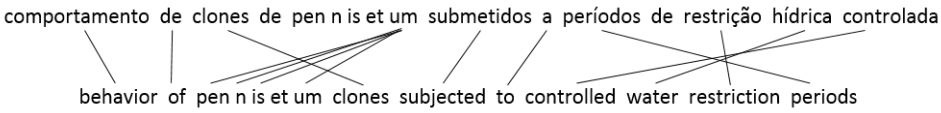
Alignment quality Tab. 7 contains example translations and subword-alignments generated from our Portuguese-English OSNMT model. Alignment links from source words consisting of multiple subwords are mapped to the final subword, visible for the words 'temperamento' in the first example and 'pennisetum' in the second one. The length of the operation sequences increases with alignment complexity as operation sequences for monotone alignments consist only of INSERT(t) and SRC_POP operations (example 1). However, even complex mappings are captured very well by OSNMT as demonstrated by the third example. Note that OSNMT can represent long-range reorderings very efficiently: the movement from 'para' in the first position to 'to' in the tenth position is simply achieved by starting the operation sequence with 'SET_MARKER to' and a JMP_BWD operation later. The first example in particular demonstrates the usefulness of such alignments as the wrong lexical choice ('abroad' rather than 'body shape') can be traced back to the source word 'exterior'.

For a qualitative assessment of the alignments produced by OSNMT we ran Giza++ to align the generated translations to the source sentences, enforced the $1{:}n$ restriction of OSNMT, and used the resulting alignments as reference for computing the alignment error rate (Och and Ney, 2003, AER). Fig. 3 shows that as training proceeds, OSNMT learns to both produce high quality translations (increasing BLEU score) and accurate alignments (decreasing AER).

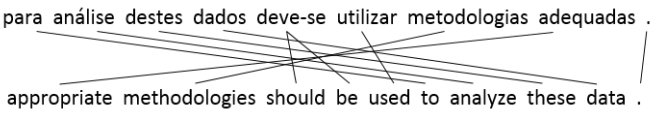
As mentioned in the introduction, a light-weight way to extract $1{:}n$ alignments from a vanilla atten-



Operation sequence: SRC_POP ab road_ SRC_POP as_ SRC_POP an_ indicator_ SRC_POP of_ SRC_POP performance_ SRC_POP and_ SRC_POP SRC_POP temper ament_ SRC_POP
Reference: the body shape as an indicative of performance and temperament



Operation sequence: behavior_ SRC_POP of_ SRC_POP SET_MARKER clones_ SRC_POP SRC_POP SRC_POP SRC_POP SRC_POP JMP_BWD pen n is et um_ SRC_POP JMP_FWD subjected_ SRC_POP to_ SRC_POP SET_MARKER periods_ SRC_POP SRC_POP JMP_BWD SET_MARKER restriction_ SRC_POP JMP_BWD SET_MARKER water_ SRC_POP JMP_BWD controlled_ SRC_POP
Reference: response of pennisetum clons to periods of controlled hidric restriction



Operation sequence: SET_MARKER to_ SRC_POP analyze_ SRC_POP these_ SRC_POP data_ SRC_POP JMP_BWD SET_MARKER should_ be_ SRC_POP used_ SRC_POP JMP_BWD SET_MARKER methodologies_ SRC_POP¿ JMP_BWD appropriate_ SRC_POP JMP_FWD JMP_FWD JMP_FWD ._ SRC_POP
Reference: to analyze these data suitable methods should be used .

Table 7: Examples of Portuguese-English translations together with their (subword-)alignments induced by the operation sequence. Alignment links from source words consisting of multiple subwords were mapped to the final subword in the training data, visible for 'temperamento' and 'pennisetum'.

Representation	Alignment extraction	AER (in %) dev	AER (in %) test
Plain	LSTM forced decoding	63.9	63.7
Plain	LSTM forced decoding with supervised attention (Liu et al., 2016, *Cross Entropy* loss)	54.9	54.7
OSNMT	OSNMT	24.2	21.5

Table 8: Comparison between OSNMT and using the attention matrix from forced decoding with a recurrent model.

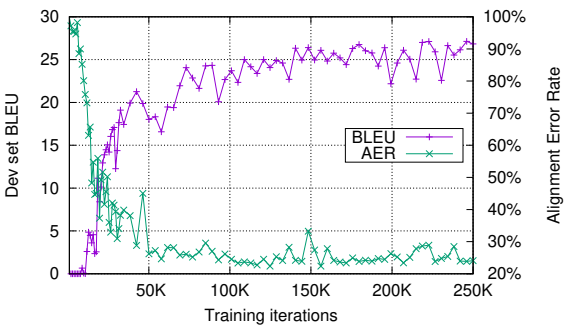
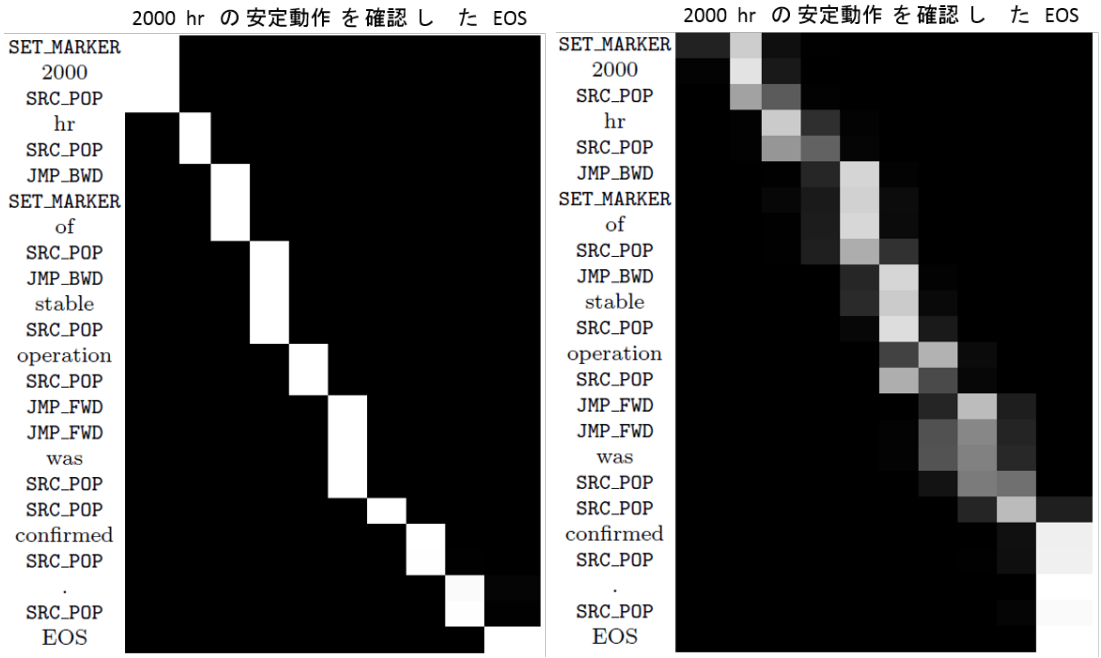


Figure 3: AER and BLEU training curves for OSNMT on the Japanese-English dev set.

tional LSTM-based seq2seq model is to take the maximum over attention weights for each target token. This is possible because, unlike the Transformer, LSTM-based models usually only have a single soft attention matrix. However, in our experiments, LSTM-based NMT was more than 4.5 BLEU points worse than the Transformer on Japanese-English. Therefore, to compare AERs under comparable BLEU scores, we used the LSTM-based models in forced decoding mode on the output of our plain text Transformer model from Tab. 6. We trained two different LSTM models: one standard model by optimizing the like-



(a) Layer 4, head 1; attending to the source side read head.

(b) Layer 2, head 3; attending to the right trigram context of the read head.

Figure 4: Encoder-decoder attention weights.

lihood of the training set, and a second one with supervised attention following Liu et al. (2016). Tab. 8 shows that the supervised attention loss of Liu et al. (2016) improves the AER of the LSTM model. However, OSNMT is able to produce much better alignments since it generates the alignment along with the translation in a single decoding run.

OSNMT sequences contain target words in source sentence order An OSNMT sequence can be seen as a sequence of target words in source sentence order, interspersed with instructions on how to put them together to form a fluent target sentence. For example, if we strip out all SRC_POP, SET_MARKER, JMP_FWD, and JMP_BWD operations in the OSNMT sequence in the second example of Tab. 7 we get:

> behavior of clones pennisetum subjected to periods restriction water controlled

The word-by-word translation back to Portugese is:

> comportamento de clones pennisetum submetidos a períodos restrição hídrica controlada

This restores the original source sentence (cf. Tab. 7) up to unaligned source words. Therefore, we can view the operations for controlling the write head (SET_MARKER, JMP_FWD, and JMP_BWD) as reordering instructions for the target words which appear in source sentence word order within the OSNMT sequence.

Role of multi-head attention In this paper, we use a standard seq2seq model (the Transformer architecture (Vaswani et al., 2017)) to generate OSNMT sequences from the source sentence. This means that our neural model is representation-agnostic: we do not explicitly incorporate the notion of read and write heads into the neural architecture. In particular, neither in training nor in decoding do we explicitly bias the Transformer's attention layers towards consistency with the alignment represented by the OSNMT sequence. Our Transformer model has 48 encoder-decoder attention matrices due to multi-head attention (8 heads in each of the 6 layers). We have found that many of these attention matrices have strong and interpretable links to the translation process represented by the OSNMT sequence. For example, Fig. 4a shows that the first head in layer 4 follows the source-side read head position very closely: at each SRC_POP operation the attention shifts by

one to the next source token. Other attention heads have learned to take other responsibilities. For instance, head 3 in layer 2 (Fig. 4b) attends to the trigram right of the source head.

5 Related Work

Explainable and interpretable machine learning is attracting more and more attention in the research community (Ribeiro et al., 2016; Doshi-Velez and Kim, 2017), particularly in the context of natural language processing (Karpathy et al., 2015; Li et al., 2016; Alvarez-Melis and Jaakkola, 2017; Ding et al., 2017; Feng et al., 2018). These approaches aim to explain (the predictions of) an existing model. In contrast, we change the target representation such that the generated sequences themselves convey important information about the translation process such as the word alignments.

Despite considerable consensus about the importance of word alignments in practice (Koehn and Knowles, 2017), e.g. to enforce constraints on the output (Hasler et al., 2018) or to preserve text formatting, introducing explicit alignment information to NMT is still an open research problem. Word alignments have been used as supervision signal for the NMT attention model (Mi et al., 2016; Chen et al., 2016; Liu et al., 2016; Alkhouli and Ney, 2017). Cohn et al. (2016) showed how to reintroduce concepts known from traditional statistical alignment models (Brown et al., 1993) like fertility and agreement over translation direction to NMT. Some approaches to simultaneous translation explicitly control for reading source tokens and writing target tokens and thereby generate monotonic alignments on the segment level (Yu et al., 2016, 2017; Gu et al., 2017). Alkhouli et al. (2016) used separate alignment and lexical models and thus were able to hypothesize explicit alignment links during decoding. While our motivation is very similar to Alkhouli et al. (2016), our approach is very different as we represent the alignment as operation sequence, and we do not use separate models for reordering and lexical translation.

The operation sequence model for SMT (Durrani et al., 2011, 2015) has been used in a number of MT evaluation systems (Durrani et al., 2014; Peter et al., 2016; Durrani et al., 2016) and for post-editing (Pal et al., 2016), often in combination with a phrase-based model. The main difference to our OSNMT is that we have adapted the set of operations for neural models and are able to use it as stand-alone system, and not on top of a phrase-based system.

Our operation sequence model has some similarities with transition-based models used in other areas of NLP (Stenetorp, 2013; Dyer et al., 2015; Aharoni and Goldberg, 2017). In particular, our `POP_SRC` operation is very similar to the *step* action of the hard alignment model of Aharoni and Goldberg (2017). However, Aharoni and Goldberg (2017) investigated monotonic alignments for morphological inflections whereas we use a larger operation/action set to model complex word reorderings in machine translation.

6 Conclusion

We have presented a way to use standard seq2seq models to generate a translation together with an alignment as linear sequence of operations. This greatly improves the interpretability of the model output as it establishes explicit alignment links between source and target tokens. However, the neural architecture we used in this paper is representation-agnostic, i.e. we did not explicitly incorporate the alignments induced by an operation sequence into the neural model. For future work we are planning to adapt the Transformer model, for example by using positional embeddings of the source read head and the target write head in the Transformer attention layers.

Acknowledgments

This work was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC grant EP/L027623/1). We thank Joanna Stadnik who produced the recurrent translation and alignment models during her 4th year project.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada. Association for Computational Linguistics.

Tamer Alkhouli, Gabriel Bretschner, Jan-Thorsten Peter, Mohammed Hethnawi, Andreas Guta, and Hermann Ney. 2016. Alignment-based neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 54–65, Berlin, Germany. Association for Computational Linguistics.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Proceedings of the Second Conference on Machine Translation*, pages 108–117, Copenhagen, Denmark. Association for Computational Linguistics.

David Alvarez-Melis and Tommi Jaakkola. 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*, pages 412–421. Association for Computational Linguistics.

Michael Auli, Adam Lopez, Hieu Hoang, and Philipp Koehn. 2009. A systematic analysis of translation model search spaces. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 224–232. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*, Toulon, France.

Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628*.

Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Agreement-based joint training for bidirectional attention-based neural machine translation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2761–2767. AAAI Press.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 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*, pages 876–885, San Diego, California. Association for Computational Linguistics.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159. Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and Stephan Vogel. 2016. QCRI machine translation systems for IWSLT 16. In *International Workshop on Spoken Language Translation. Seattle, WA, USA*.

Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. 2014. Edinburgh's phrase-based machine translation systems for WMT-14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 97–104, Baltimore, Maryland, USA. Association for Computational Linguistics.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1045–1054, Portland, Oregon, USA. Association for Computational Linguistics.

Nadir Durrani, Helmut Schmid, Alexander Fraser, Philipp Koehn, and Hinrich Schütze. 2015. The operation sequence model—combining n-gram-based and phrase-based statistical machine translation. *Computational Linguistics*, 41(2):157–186.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. 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)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *ArXiv e-prints*.

Hamidreza Ghader and Christof Monz. 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)*, pages 30–39. Asian Federation of Natural Language Processing.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 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, pages 1053–1062. Association for Computational Linguistics.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. 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*, pages 681–691. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

I. Dan Melamed. 2003. Multitext grammars and synchronous parsers. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

I. Dan Melamed, Giorgio Satta, and Benjamin Wellington. 2004. Generalized multitext grammars. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, Texas. Association for Computational Linguistics.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *LREC*, pages 2204–2208, Portoroz, Slovenia.

Mariana L Neves and Aurélie Névéol. 2016. The Scielo corpus: a parallel corpus of scientific publications for biomedicine. In *LREC*, pages 2942–2948, Portoroz, Slovenia.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Santanu Pal, Marcos Zampieri, and Josef van Genabith. 2016. USAAR: An operation sequential model for automatic statistical post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 759–763, Berlin, Germany. Association for Computational Linguistics.

Jan-Thorsten Peter, Andreas Guta, Nick Rossenbach, Miguel Graça, and Hermann Ney. 2016. The RWTH Aachen machine translation system for IWSLT 2016. In *International Workshop on Spoken Language Translation. Seattle, WA, USA*.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 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*, pages 97–101. Association for Computational Linguistics.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 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*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Felix Stahlberg, Eva Hasler, Danielle Saunders, and Bill Byrne. 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*, pages 25–30. Association for Computational Linguistics. Full documentation available at http://ucam-smt.github.io/sgnmt/html/.

Pontus Stenetorp. 2013. Transition-based dependency parsing using recursive neural networks. In *NIPS Workshop on Deep Learning*. Citeseer.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

L Yu, P Blunsom, C Dyer, E Grefenstette, and T Kocisky. 2017. The neural noisy channel. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France. Computational and Biological Learning Society.

Lei Yu, Jan Buys, and Phil Blunsom. 2016. Online segment to segment neural transduction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1307–1316, Austin, Texas. Association for Computational Linguistics.