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.,
The final translation in plain text can be obtained if the read head is at the end of the source sentence. ‘2000’. The translation process terminates when to the marker $X$ and ‘hr’, in time step 6 the model jumps back to the initial marker (highlighted). The model begins with 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 $G$ which generates 1.) the source sentence, 2.) the target sentence, and 3.) the position of the target side write head. The start symbol of $G$ is

$$[(S), (X_1), (P_1)]^T$$

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 $G$ as

$$[(\alpha_1), (\alpha_2), (\alpha_3)]^T \rightarrow [(\beta_1), (\beta_2), (\beta_3)]^T$$

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 $\alpha_i$ are the rewriting rules for each of
<table>
<thead>
<tr>
<th>Operation</th>
<th>Source sentence</th>
<th>Target sentence (compiled)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 SET_MARKER</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_1$</td>
</tr>
<tr>
<td>2 2000</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_2$ $X_1$</td>
</tr>
<tr>
<td>3 POP_SRC</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_2$ 2000 $X_1$</td>
</tr>
<tr>
<td>4 hr</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>5 POP_SRC</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>6 JMP_BWD</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>7 SET_MARKER</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_3$ $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>8 of</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>9 POP_SRC</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>10 JMP_BWD</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>$X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>11 stable</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable $X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>12 operation</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>13 POP_SRC</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>14 JMP_FWD</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>15 JMP_FWD</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr $X_1$</td>
</tr>
<tr>
<td>16 was</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr was $X_1$</td>
</tr>
<tr>
<td>17 POP_SRC</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr was $X_1$</td>
</tr>
<tr>
<td>18 confirmed</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr was confirmed $X_1$</td>
</tr>
<tr>
<td>21 POP_SRC</td>
<td>2000 hr の 安定 動作 を 確認 した</td>
<td>stable operation $X_3$ of $X_2$ 2000 hr was confirmed $X_1$</td>
</tr>
</tbody>
</table>

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.

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

POP_SRC : $\forall s \in \mathcal{V}_{src} : \left[ (S) \right] \rightarrow \left[ (sS) \right] \quad (3)$

where $\mathcal{V}_{src}$ is the source language vocabulary. A jump from marker $X_i$ to $X_j$ is realized by replac-

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

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

Figure 2: Target-side tree representation of the operation sequence in Tab. 1.
Tab. 2 illustrates that there is a 1:1 correspondence between a derivation in $G$ and an OSNMT operation sequence. The target-side derivation (the second component in $G$) 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 $G$ is context-free in the second component, our scoring model for $G$ 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) is inspired by the original OSM for SMT (Durrani et al., 2011) as it also represents the translation process as a 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 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.
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.

## 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.

Table 3: Training set sizes.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Language pair</th>
<th># Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scielo</td>
<td>Spanish-English</td>
<td>587K</td>
</tr>
<tr>
<td>Scielo</td>
<td>Portuguese-English</td>
<td>513K</td>
</tr>
<tr>
<td>WAT</td>
<td>Japanese-English</td>
<td>1M</td>
</tr>
</tbody>
</table>

³We follow the transformer_base configuration and use 6 layers, 512 hidden units, and 8 attention heads in both the encoder and decoder.

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

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

---

**Algorithm 1 Align2OSNMT(a, x, y)**

```python
1: holes ← \{(0, ∞)\}
2: ops ← ∅ \{Initialize with empty list\}
3: head ← 0
4: for i ← 1 to |x| do
5:   for all j ∈ \{j|a_j = i\} do
6:     hole_idx ← holes.find(j)
7:     d ← hole_idx - head
8:   if d < 0 then
9:     ops.extend(JMP_BWD.repeat(−d))
10: end if
11: if d > 0 then
12:   ops.extend(JMP_FWD.repeat(d))
13: end if
14: head ← hole_idx
15: (s, t) ← holes[head]
16: if s ≠ j then
17:   holes.append((s, j − 1))
18:   head ← head + 1
19: ops.append(SEM_MARKER)
20: end if
21: ops.append(\(y_j\))
22: holes[head] ← (j + 1, t)
23: end for
24: ops.append(SRC_POP)
25: end for
26: return ops
```

---

Note: The table and algorithm are extracted and formatted into a readable format. The text is a detailed explanation of the training process for a multilingual Transformer model, emphasizing the differences between plain text NMT and OSNMT. The algorithm for converting a bilingual corpus into an OSNMT representation is shown, along with the key steps for generating operation sequences and aligning the source and target sentences. The results section outlines the evaluation on three language pairs, highlighting the use of jointly trained models and the influence of beam search parameters on translation performance.
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 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.

Table 4: Generating training alignments on the subword level.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU es-en</th>
<th>BLEU pt-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Align on subword level</td>
<td>36.7</td>
<td>38.1</td>
</tr>
<tr>
<td>Convert word level alignments</td>
<td>37.1</td>
<td>38.4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid</td>
<td>92.49%</td>
</tr>
<tr>
<td>Not enough SRC_POP</td>
<td>7.28%</td>
</tr>
<tr>
<td>Too many SRC_POP</td>
<td>0.22%</td>
</tr>
<tr>
<td>Write head out of range</td>
<td>0.06%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Representation</th>
<th>BLEU es-en</th>
<th>BLEU pt-en</th>
<th>BLEU ja-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>37.6</td>
<td>37.5</td>
<td>28.3</td>
</tr>
<tr>
<td>OSNMT</td>
<td>37.1</td>
<td>38.4</td>
<td>28.1</td>
</tr>
</tbody>
</table>

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-
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’.

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.
likelihood 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 Portuguese 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...
| Page 1 |

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


