# Jointly Learning to Align and Translate with Transformer Models

**Sarthak Garg** **Stephan Peitz** **Udhyakumar Nallasamy** **Matthias Paulik**
Apple Inc.
{sarthak_garg, speitz, udhay, mpaulik}@apple.com

## Abstract

The state of the art in machine translation (MT) is governed by neural approaches, which typically provide superior translation accuracy over statistical approaches. However, on the closely related task of word alignment, traditional statistical word alignment models often remain the go-to solution. In this paper, we present an approach to train a Transformer model to produce both accurate translations and alignments. We extract discrete alignments from the attention probabilities learnt during regular neural machine translation model training and leverage them in a multi-task framework to optimize towards translation and alignment objectives. We demonstrate that our approach produces competitive results compared to GIZA++ trained IBM alignment models without sacrificing translation accuracy and outperforms previous attempts on Transformer model based word alignment. Finally, by incorporating IBM model alignments into our multi-task training, we report significantly better alignment accuracies compared to GIZA++ on three publicly available data sets. Our implementation has been open-sourced[1].

## 1 Introduction

Neural machine translation (NMT) constitutes the state of the art in MT, with the Transformer model architecture (Vaswani et al., 2017) beating other neural architectures in competitive MT evaluations. The attention mechanism used in NMT models was motivated by the need to model word alignments, however it is now well known that the attention probabilities can differ significantly from word alignments in the traditional sense (Koehn and Knowles, 2017), since attending to the context words rather than the aligned source words might be helpful for translation. The presence of multi-layer, multi-head attention mechanisms in the Transformer model further complicate interpreting the attention probabilities and extracting high quality discrete alignments from them.

Finding source to target word alignments has many applications in the context of MT. A straightforward application of word alignments is to generate bilingual lexica from parallel corpora. Word alignments have also been used for external dictionary assisted translation (Chatterjee et al., 2017; Alkhouli et al., 2018; Arthur et al., 2016) to improve translation of low frequency words or to comply with certain terminology guidelines. Documents and webpages often contain word annotations such as formatting styles and hyperlinks, which need to be preserved in the translation. In such cases, word alignments can be used to transfer these annotations from the source sentence to its translation. In user facing translation services, providing word alignments as additional information to the users might improve their trust and confidence, and also help them to diagnose problems such as under-translation (Tu et al., 2016).

In this work, we introduce an approach that teaches Transformer models to produce translations and interpretable alignments simultaneously:

- We use a multi-task loss function combining negative log likelihood (NLL) loss used in regular NMT model training and an alignment loss supervising one attention head to learn alignments (Section 4.2).
- Conditioning on past target context is essential for maintaining the auto-regressive property for translation but can be limiting for alignment. We alleviate this problem by conditioning the different components of our multi-task objective on different amounts of context (Section 4.3).

[1]Code can be found at https://github.com/pytorch/fairseq/pull/1095



*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4453–4462, Hong Kong, China, November 3–7, 2019. 

- We demonstrate that the system can be supervised using seed alignments obtained by carefully averaging the attention probabilities of a regular NMT model (Section 4.1) or alignments obtained from statistical alignment tools (Section 4.4)

We show that our model outperforms previous neural approaches (Peter et al., 2017; Zenkel et al., 2019) and statistical alignment models (Och and Ney, 2003) in terms of alignment accuracy without suffering any degradation of translation accuracy.

## 2 Preliminaries

### 2.1 Word Alignment Task

Given a sentence $f_1^J = f_1, \ldots, f_j, \ldots f_J$ in the source language *and* its translation $e_1^I = e_1, \ldots, e_i, \ldots e_I$ in the target language, an alignment $\mathcal{A}$ is defined as a subset of the Cartesian product of the word positions (Och and Ney, 2003).

$$\mathcal{A} \subseteq \{(j, i) : j = 1, \ldots, J; i = 1, \ldots, I\} \quad (1)$$

The word alignment task aims to find a discrete alignment representing a many-to-many mapping from the source words to their corresponding translations in the target sentence.

### 2.2 Transformer Model

The Transformer model (Vaswani et al., 2017) is an encoder-decoder model that only relies on attention for computing the contextual representations for source and target sentences. Both the encoder and decoder are composed of multiple layers, each of which includes a multi-head self-attention and a feed-forward sub-layer. Layers in the decoder additionally apply a multi-head encoder-decoder attention between the self-attention and the feed-forward sub-layers. To maintain the auto-regressive property, the self-attention sub-layer in the decoder attends to the representations of only the *past* tokens computed by the lower layer.

In this work, we will be focusing on guiding the encoder-decoder attention sub-layer in the decoder. Let $d_{emb}, d_k, d_v, N$ denote the embedding dimension, dimensions of the key and value projections and number of heads, respectively. As described in Vaswani et al. (2017), for this sub-layer, the output of the previous decoder layer corresponding to the $i^{th}$ target token is used as a query vector $\mathbf{q}^i \in \mathbb{R}^{1 \times d_{emb}}$ and the encoder output for all the source tokens are packed together as the value $V \in \mathbb{R}^{J \times d_{emb}}$ and key $K \in \mathbb{R}^{J \times d_{emb}}$ matrices. To compute the output $\mathcal{M}(\mathbf{q}^i, K, V)$, $N$ heads first project the query vector and the key and value matrices into different subspaces, compute attention in their own subspaces, aggregate their outputs and project back to the original space:

$$\tilde{\mathbf{q}}_n^i = \mathbf{q}^i W_n^Q, \tilde{K}_n = K W_n^K, \tilde{V}_n = V W_n^V \quad (2)$$

$$H_n^i = \text{Attention}(\tilde{\mathbf{q}}_n^i, \tilde{K}_n, \tilde{V}_n) \quad (3)$$

$$\mathcal{M}(\mathbf{q}^i, K, V) = \text{Concat}(H_1^i, \ldots, H_N^i) W^O, \quad (4)$$

where the projection matrices $W_n^Q, W_n^K, W_n^V$ and $W^O$ are learnt parameters of the $n^{th}$ head. Each head employs a scaled dot-product attention:

$$\text{Attention}(\tilde{\mathbf{q}}_n^i, \tilde{K}_n, \tilde{V}_n) = \mathbf{a}_n^i \tilde{V}_n, \quad (5)$$

$$\text{where } \mathbf{a}_n^i = \text{softmax}(\frac{\tilde{\mathbf{q}}_n^i \tilde{K}_n^T}{\sqrt{d_k}}). \quad (6)$$

The vector $\mathbf{a}_n^i \in \mathbb{R}^{1 \times J}$ denotes the attention probabilities for the $i^{th}$ target token over all the source tokens, computed by the $n^{th}$ attention head. For any particular head, an attention matrix $A_{I \times J}$ can be constructed by grouping together the vectors $\mathbf{a}_n^i$ corresponding to all the target tokens. In the following sections, we analyze the quality of alignments that can be extracted from these attention matrices $A_{I \times J}$ and describe how they can be effectively supervised to learn word alignments.

## 3 Baseline Methods

A common baseline approach to extract word alignments from a regular NMT trained Transformer, is to average over all attention matrices $A_{I \times J}$ computed across all layers and heads. The resulting matrix gives a probability distribution over all source tokens for each target token. This distribution is then converted to a discrete alignment by aligning each target word to the corresponding source word with the highest attention probability.

Peter et al. (2017) guide the attention probabilities to be close to the alignments obtained from statistical MT toolkits by imposing an additional loss based on the distance between the alignment and attention distributions. They get improvements in alignment accuracy over previous works based on guided alignment training by feeding the *current* target word to the attention module, providing it more context about the target sentence.

Zenkel et al. (2019) proposed an method that does not rely on alignments from external toolkits for training. They instead add an extra attention layer on top of the Transformer architecture and directly optimize its activations towards predicting the given target word.

All the above methods involve training models for both the directions to get bidirectional alignments. These bidirectional alignments are then merged using the `grow diagonal` heuristic (Koehn et al., 2005).

## 4 Proposed Method

### 4.1 Averaging Layer-wise Attention Scores

The attention heads in a single layer are symmetrical, but the different layers themselves can learn drastically different alignments. To better understand the behavior of the encoder-decoder attention learnt at different layers, we average the attention matrices computed across all heads within each layer and evaluate the obtained alignments. We show that the attention probabilities from the penultimate layer naturally tend to learn alignments and provide significantly better results compared to naively averaging across all layers (cf. Section 5.3). For the rest of the paper, we refer to the former method as the layer average baseline.

### 4.2 Multi-task Learning

Translation and alignment tasks are very closely related. NMT models with attention (Bahdanau et al., 2015) have also shown to learn alignments in the intermediate attention layer. A neural model receiving supervision from given translations *and* given alignments can therefore benefit from multi-task learning by exploiting the correlations between these two tasks.

Annotating word alignments is a laborious and expensive task, but the layer average baseline described in Section 4.1 is able to generate reasonably good alignments in an unsupervised manner. We thus use the alignments generated by the layer average baseline as labels for supervising our model. We first convert the alignments into a probability distribution over source tokens for every target token. Let $G_{I\times J}$ denote a 0-1 matrix such that $G_{i,j} = 1$ if the $j^{th}$ source token is aligned to the $i^{th}$ target token. We simply normalize the rows of matrix $G$ corresponding to target tokens that are aligned to at least one source token to get a matrix $G^p$. As described in Section 2.2, the Transformer model computes multiple attention probability distributions over source tokens for every target token across different heads and layers of the network. Since we observed that the attention probabilities from the penultimate layer most naturally tend to learn alignments (Section 5.3), we arbitrarily select one head from the penultimate layer (subsequently referred to as the alignment head) and supervise its attention probability distribution to be close to the labeled alignment distribution ($G^p$). Let $A_{I\times J}$ denote the attention matrix computed by the alignment head. For every target word $i$, we minimize the Kullback-Leibler divergence between $G^p_i$ and $A_i$ which is equivalent to optimizing the following cross-entropy loss $\mathcal{L}_a$

$$\mathcal{L}_a(A) = -\frac{1}{I}\sum_{i=1}^{I}\sum_{j=1}^{J} G^p_{i,j}\log(A_{i,j}). \quad (7)$$

The motivation behind supervising one head is that it gives the model the flexibility to either use the representation computed by the alignment head, or depend more on the representations computed by other heads. We train our model to minimize $\mathcal{L}_a$ in conjunction with the standard NLL translation loss $\mathcal{L}_t$. The overall loss $\mathcal{L}$ is:

$$\mathcal{L} = \mathcal{L}_t + \lambda\mathcal{L}_a(A), \quad (8)$$

where $\lambda$ is a hyperparameter.

### 4.3 Providing Full Target Context

The Transformer decoder computes the probability of the next target token conditioned on the past target tokens and all source tokens. This is implemented by masking the self attention probabilities, i.e. while computing the representation for the $i^{th}$ target token, the decoder can only self-attend to the representations of $\{1, 2 \ldots i-1\}$ tokens from the previous layer. This auto-regressive behavior of the decoder is crucial for the model to represent a valid probability distribution over the target sentence. However, conditioning on just the past target tokens is limiting for the alignment task. As described in Section 4.2, the alignment head is trained to model the alignment distribution for the $i^{th}$ target token given only the past target tokens and all source tokens. Since the alignment head does not know the identity of the next target token, it becomes difficult for it to learn this token's alignment to the source tokens. Previous work has also identified this problem and alleviate

it by feeding the target token to be aligned as an input to the module computing the alignment (Peter et al., 2017), or forcing the module to predict the target token (Zenkel et al., 2019) or its properties, e.g. POS tags (Li et al., 2018). Feeding the next target token assumes that we know it in advance and thus calls for separate translation and alignment models. Forcing the alignment module to predict target token's properties helps but still passes the information of the target token in an indirect manner. We overcome these limitations by conditioning the two components of our loss function on different amounts of context. The NLL loss $\mathcal{L}_t$ is conditioned on the past target tokens to preserve the auto-regressive property:

$$\mathcal{L}_t = -\frac{1}{I}\sum_{i=1}^{I}\log(p(e_i|f_1^J, e_1^{i-1})). \quad (9)$$

However, the alignment loss $\mathcal{L}_a^{'}$ is now conditioned on the whole target sentence:

$$\mathcal{L}_a^{'} = \mathcal{L}_a(A|f_1^J, e_1^I). \quad (10)$$

This is implemented by executing two forward passes of the decoder model, one with the masking of the future target tokens for computing the NLL loss $\mathcal{L}_t$ and the other one with no masking for computing the alignment loss $\mathcal{L}_a^{'}$from the alignment head. Although this formulation forces the network to learn representations adapting to both full and partial target context, Section 5.5 shows that this approach does not degrade the translation quality while improving the alignment accuracy.

### 4.4 Alignment Training Data

Our method described so far does not rely on alignments from external statistical toolkits but performs self-training on alignments extracted from the layer average baseline. However, GIZA++ provides a robust method to compute accurate alignments. If achieving better alignment accuracy is paramount, then our multi-task framework can also leverage alignments from GIZA++ to produce even better alignment accuracy (Section 5.4). In this setting we use the GIZA++ alignments as labels instead of those obtained from the layer average baseline for supervising the alignment head.

## 5 Experiments

### 5.1 Setup

Our experiments show that our proposed approach is able to achieve state-of-the-art results in terms of alignment *and* maintain the same translation performance. In the following, we describe two setups to compare with previously established state-of-the-art results.

For all setups and models used in this work, we learn a joint source and target Byte-Pair-Encoding (BPE, Sennrich et al. (2016)) with 32k merge operations. We observe that even for statistical alignment models sub-word units are beneficial. To convert the alignments from sub-word-level back to word-level, we consider each target word as being aligned to a source word if an alignment between any of the target sub-words and source sub-words exists.

The alignment quality is evaluated by using the alignment error rate (AER) introduced in (Och and Ney, 2000). Significance of the differences in AER between two models is tested using a two-sided Wilcoxon signed-rank test ($\alpha = 0.1\%$).

#### 5.1.1 Alignment Task

The purpose of the this task is to fairly compare with state-of-the-art results in terms of alignment quality and perform a hyperparameter search. We use the same experimental setup as described in (Zenkel et al., 2019). The authors provide pre-processing and scoring scripts[2] for three different datasets: Romanian→English, English→French and German→English. Training data and test data for Romanian→English and English→French are provided by the NAACL'03 *Building and Using Parallel Texts* word alignment shared task[3] (Mihalcea and Pedersen, 2003). The Romanian→English training data are augmented by the Europarl v8 corpus increasing the amount of parallel sentences from 49k to 0.4M. For German→English we use the Europarl v7 corpus as training data and the gold alignments[4] provided by Vilar et al. (2006). The reference alignments were created by randomly selecting a subset of the Europarl v7 corpus and manually annotating them following the guidelines suggested in (Och

[2] https://github.com/lilt/alignment-scripts

[3] http://web.eecs.umich.edu/~mihalcea/wpt/index.html#resources

[4] https://www-i6.informatik.rwth-aachen.de/goldAlignment/

and Ney, 2003). Data statistics are shown in Table 1.

Table 1: Number of sentences for three datasets: German→English (DeEn), Romanian→English (RoEn) and English→French (EnFr). The datasets include training data and test data with gold alignments.

| | DeEn | RoEn | EnFr |
|---|---|---|---|
| training | 1.9M | 0.5k | 1.1M |
| test | 508 | 248 | 447 |

In all experiments for this task, we employ the `base` transformer configuration with an embedding size of 512, 6 encoder and decoder layers, 8 attention heads, shared input and output embeddings (Press and Wolf, 2017), the standard `relu` activation function and sinusoidal positional embedding. The total number of parameters is 60M. We train with a batch size of 2000 tokens on 8 Volta GPUs and use the validation translation loss for early stopping. Furthermore, we use Adam optimizer (Loshchilov and Hutter, 2019) with a learning rate of 3e-4, $\beta_1 = 0.9, \beta_2 = 0.98$, learning rate warmup over the first 4000 steps and inverse square root as learning rate scheduler. The dropout probability is set to 0.1. Additionally, we apply label smoothing with a factor of 0.1. To conveniently extract word alignments for both translation directions, we train bidirectional models, i.e. our models are able to translate and align from Romanian to English and vice versa.

#### 5.1.2 Align and Translate Task

The second setup is based on the WMT'18 English-German news translation task (Bojar et al., 2018). We apply the same corpus selection for bilingual data and model architecture as suggested by Edunov et al. (2018). However, we slightly modify the preprocessing pipeline to be able to evaluate the alignment quality against the gold alignments provided by Vilar et al. (2006). We use all available bilingual data (Europarl v7, Common Crawl corpus, News Commentary v13 and Rapid corpus of EU press releases) excluding the ParalCrawl corpus. We remove sentences longer than 100 words and sentence pairs with a source/target length ratio exceeding 1.5. This results in 5.2M parallel sentences. We apply the Moses tokenizer (Koehn et al., 2007) without aggressive hyphen splitting and without performing HTML escaping of apostrophes and quotes. Furthermore, we do not normalize punctuation marks. We use newstest2012 as validation and newstest2014 as test set.

To achieve state-of-the-art translation results, all models in this setup are trained unidirectional and we change to the `big` transformer configuration with an embedding size of 1024 and 16 attention heads. The total number of parameters is 213M. We train the layer average baseline with a batch size of 7168 tokens on 64 Volta GPUs for 30k updates and apply a learning rate of 1e-3, $\beta_1 = 0.9, \beta_2 = 0.98$. The dropout probability is set to 0.3. All other hyperparameters are as described in the previous section. Since training the multi-task models consumes more memory, we need to half the batch size, increase the number of updates accordingly and adapt the learning rate to 7e-4. We average over the last 10 checkpoints and run inference with a beam size of 5.

To fairly compare against state-of-the-art translation setups, we compute BLEU (Papineni et al., 2002) with `sacreBLEU` (Post, 2018).

### 5.2 Statistical Baseline

For both setups, the statistical alignment models are computed with the multi-threaded version of the GIZA++ toolkit[5] implemented by Gao and Vogel (2008). GIZA++ estimates IBM1-5 models and a first-order hidden Markov model (HMM) as introduced in (Brown et al., 1993) and (Vogel et al., 1996), respectively. In particular, we perform 5 iterations of IBM1, HMM, IBM3 and IBM4. Furthermore, the alignment models are trained in both translation directions and symmetrized by employing the `grow-diagonal` heuristic (Koehn et al., 2005). We use the resulting word alignments to supervise the alignment loss for the method described in Section 4.4.

### 5.3 Averaging Attention Results

For our experiments, we use the data and Transformer model setup described in Section 5.1.1. We perform the evaluation of alignments obtained by layer wise averaging of attention probabilities as described in Section 4.1. As shown in Table 2, all three language pairs exhibit a very similar pattern, wherein the attentions do not seem to learn meaningful alignments in the initial layers and show a remarkable improvement in the higher layers. This indicates that the initial layers are fo-

[5] https://github.com/moses-smt/mgiza/

Table 2: AER [%] per layer for all three language pairs: German→English (DeEn), Romanian→English (RoEn) and English→French (EnFr).

| Layer | DeEn | RoEn | EnFr |
|---|---|---|---|
| 1 (bottom) | 90.0 | 92.9 | 80.7 |
| 2 | 91.0 | 93.6 | 81.7 |
| 3 | 94.5 | 92.0 | 73.4 |
| 4 | 41.2 | 37.5 | 20.5 |
| 5 | **32.6** | **33.4** | **17.0** |
| 6 (top) | 56.3 | 48.4 | 37.9 |
| average | 55.8 | 38.6 | 23.2 |

cusing more on learning good representations of the sentence generated by the decoder so far by self attention. Once good contextual representations are learnt, the higher layers fetch the relevant representations from the encoder's output via the encoder-decoder attention. However, interestingly the penultimate layer outperforms the final layer suggesting the final layer uses the alignment-based features in the penultimate layer to derive its own representation.

### 5.4 Alignment Task Results

Table 3 compares the performance of our methods against statistical baselines and previous neural approaches. The layer average baseline provides relatively weak alignments, which are used for training our *multi-task* model. The improvement of the multi-task approach over the layer average baseline suggests that learning to translate helps produce better alignments as well. However still the multi-task approach falls short of the statistical and neural baselines, which have a strong advantage of having access to the full/partial target context. Exposing our model to the full target context gives the largest gains in terms of AER. Note that *full context* results are directly comparable to Zenkel et al. (2019) since both approaches do not leverage external knowledge from statistical models. We suspect that we are able to outperform Zenkel et al. (2019) because we provide the full target context instead of only the to-be aligned target word. Finally, by supervising our model on the alignments obtained from GIZA++ (GIZA++ *supervised*) rather than layer average baseline, we outperform GIZA++ and Peter et al. (2017).

We tuned the alignment loss weight $\lambda$ (Equation 8) using grid search on the German→English dataset. We achieve the best results with $\lambda = 0.05$.

Table 3: Results on the alignment task (in AER [%]). ‡Difference in AER w.r.t. GIZA++ (BPE-based) is statistically significant ($p<0.001$).

| Model | DeEn | RoEn | EnFr |
|---|---|---|---|
| GIZA++ (word-based) | 21.4 | 27.9 | 5.9 |
| GIZA++ (BPE-based) | 18.9 | 27.0 | 5.5 |
| Layer average baseline | 32.6 | 33.4 | 17.0 |
| Multi-task | 25.4 | 30.7 | 12.6 |
| + full-context | 20.2 | 26.0 | 7.7 |
| ++ GIZA++ supervised | **16.0**‡ | **23.1**‡ | **4.6**‡ |
| Peter et al. (2017) | 19.0 | - | - |
| Zenkel et al. (2019) | 21.2 | 27.6 | 10.0 |

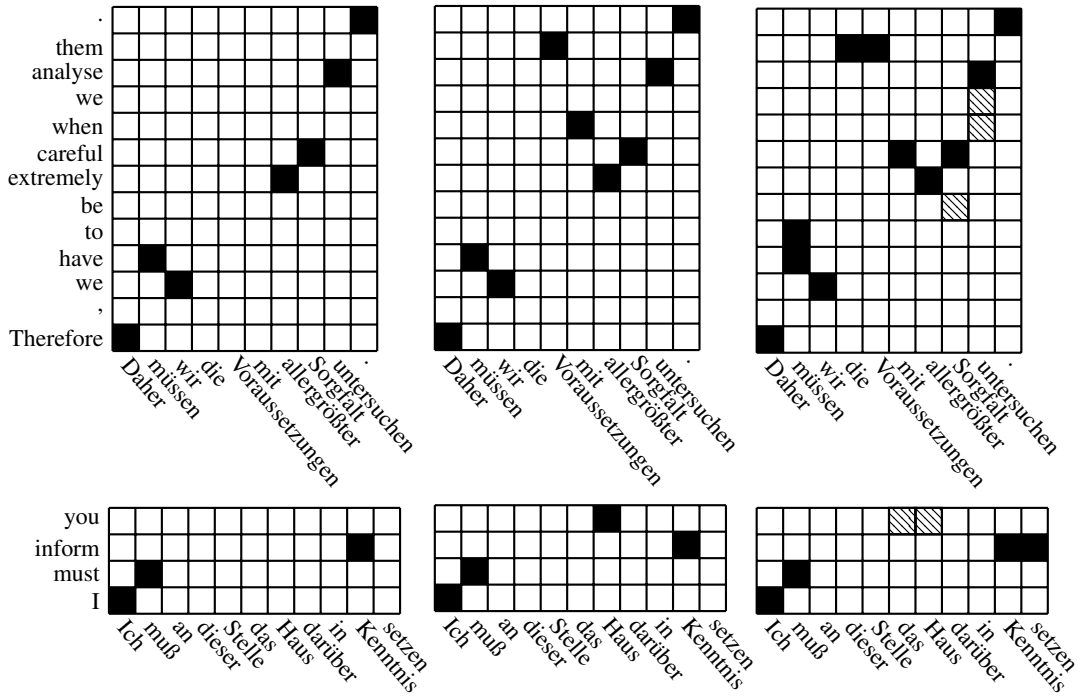
### 5.5 Align and Translate Task Results

For fair comparison of our approach to the state-of-the-art translation models, we use the setup described in Section 5.1.2. Table 4 summarizes the results on alignment and translation tasks. The layer average baseline is based on regular NMT model training, therefore ideally it should achieve the same BLEU as Edunov et al. (2018), however we see a small drop of 0.3 BLEU points in practice which could be caused by the slightly different preprocessing procedure (cf. Section 5.1.2, no aggressive hyphen splitting/no punctuation normalization). The layer average baseline performs poorly in terms of the AER. The Precision and Recall results for the layer average baseline demonstrate the effectiveness of symmetrization. Symmetrization removes a majority of incorrect alignments and gives a high precision (94.2%) but low recall (29.6%). The high precision of the layer average baseline ensures that the multi-task model receives correct alignments for supervision, enabling it to get large improvements in AER over the layer average baseline.

Similar to the trend observed in 5.4, providing full target sentence context in the decoder helps the model to improve further and perform comparably to GIZA++. Lastly, supervision with GIZA++ gives the best AER and significantly outperforms GIZA++. The improvements in alignment quality and no degradation in BLEU compared to the layer average baseline shows the effectiveness of the proposed multi-task approach.

Table 4: Results on the align and translate task. Alignment quality is reported in AER, translation quality in BLEU. †baseline (without back-translation) `sacreBLEU` results were provided in https://github.com/pytorch/fairseq/issues/506#issuecomment-464411433. ‡Difference in AER w.r.t. GIZA++ (BPE-based) is statistically significant ($p<0.001$)

| Model | AER[%] (Precision[%], Recall[%]) DeEn | EnDe | Symmetrized | BLEU[%] DeEn | EnDe |
|---|---|---|---|---|---|
| GIZA++ (word-based) | 21.7 (85.4, 72.1) | 24.0 (85.8, 68.2) | 22.2 (93.5, 66.5) | - | - |
| GIZA++ (BPE-based) | 19.0 (89.1, 74.2) | 21.3 (86.8, 71.9) | 19.6 (93.2, 70.6) | - | - |
| Layer average baseline | 66.8 (32.0, 34.6) | 66.5 (32.5, 34.7) | 54.8 (94.2, 29.6) | 33.1 | 28.7 |
| Multi-task | 31.1 (67.2, 70.7) | 32.2 (66.6, 69.1) | 25.8 (88.1, 63.8) | 33.1 | 28.5 |
| + full-context | 21.2 (76.9, 80.9) | 23.5 (75.0, 78.0) | 19.5 (89.5, 72.9) | 33.2 | 28.5 |
| ++ GIZA++ supervised | **17.5**‡ (80.5, 84.7) | **19.8**‡ (78.8, 81.7) | **16.4**‡ (89.6, 78.2) | 33.1 | 28.8 |
| Edunov et al. (2018)† | - | - | - | - | 29.0 |




(a) GIZA++ (b) Our Model (c) Reference

Figure 1: Two examples from the German→English alignment test set. Alignments in (a) show the output from GIZA++ and (b) from our model (*Multi-task* with full-context and GIZA++ supervised). Gold Alignments in shown in (c). Black squares and hatched squares in the reference represent *sure* and *possible* alignments, respectively.

## 6 Analysis

To further investigate why our proposed approach is superior to GIZA++ in terms of AER, we analyze the generated word alignments of both models. We observe that our model tends to align pronouns (e.g. *you* or *them*) with regular nouns (e.g. objects or subjects). Given the gold alignments, it seems that these alignment links are correct or at least possible (Och and Ney (2003) provided annotators two options to specify alignments: *sure* and *possible* for unambiguous and ambiguous alignments respectively). Figure 1 shows two examples from the German→English alignment test set. In the first example, our model correctly aligns *them* with *Voraussetzungen (criteria)*. The German parliament speaker indeed mentioned *Verfahrensvoraussetzungen (procedural criteria)* in one of the preceding sentences and refers later to them by using the term *Voraussetzungen (criteria)*. In the second example, the pronoun *you* is correctly aligned to the noun *Haus*

*(house)* which is just another way to address the audience in the European parliament. Both alignment links are not generated by GIZA++. This could be related to fact that a statistical model is based on counting co-occurrences. We speculate that to generate such alignment links, a model needs to be able to encode contextual information. Experimental results in (Tang et al., 2018) suggest that NMT models learn to encode contextual information, which seems to be necessary for word sense disambiguation. Since pronouns can be ambiguous references, we assume that both problems are closely related and therefore believe that the ability to encode contextual information may be beneficial for generating word alignments.

From our experiments on the WMT'18 dataset, we observe that the alignment quality of the layer average baseline is quite low (cf. Table 4). To further investigate this, we plot the test AER and the validation NLL loss per epoch (Figure 2). The graph shows that the lowest AER of 42.7% is already reached in the fifth epoch. This suggests that

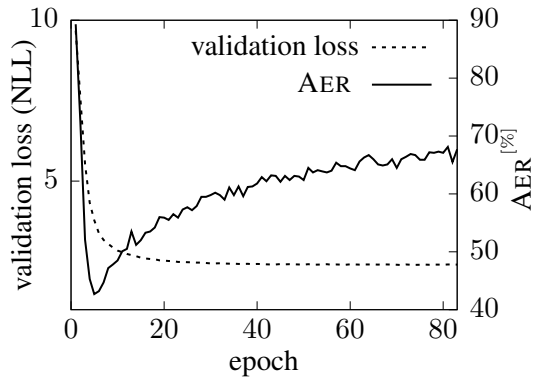



Figure 2: Test AER and validation loss (NLL) per epoch on the WMT'18 English→German task.

picking an earlier checkpoint for generating word alignments could be beneficial for better supervision. Unfortunately, an alignment validation set does not exist for this task.

## 7 Related Work

Leveraging alignments obtained from statistical MT toolkits to guide NMT attention mechanisms has been explored in the past. Mi et al. (2016), Chen et al. (2016), Liu et al. (2016) and Alkhouli and Ney (2017) supervise the attention mechanisms of recurrent models (Bahdanau et al., 2015) in this way. Our multi-task framework is inspired by these publications. However, we examine its effect on the Transformer model (Vaswani et al., 2017), which provides state-of-the-art results on several translation benchmarks. Previous works report significant gains in translation accuracy in low resource settings, however gains remain modest given larger amounts of parallel data (millions of sentences). These approaches also fail to achieve significantly better alignment accuracy than the statistical MT toolkits. Peter et al. (2017) and Li et al. (2018) improve upon the previous works in terms of alignment accuracy by providing an alignment module with additional information about the to-be-aligned target word. Expanding on this idea, we propose to leverage the full target sentence context leading to AER improvements. Zenkel et al. (2019) presents an approach that eliminates the reliance on statistical word aligners by instead by directly optimizing the attention activations for predicting the target word. We empirically compare our approach of obtaining high quality alignments without the need of statistical word aligners to Zenkel et al. (2019).

Augmenting the task objective with linguistic information, such as word alignments, also has had applications beyond MT. Strubell et al. (2018) showed that adding linguistic information from parse trees into one of the attention heads of the transformer model can help in the semantic role labeling. Inspired by Strubell et al. (2018), we inject the alignment information through one of the attention heads for the translation task instead.

As a by-product of developing our model, we present a simple way to quantitatively evaluate and analyze the quality of attention probabilities learnt by different parts of the Transformer model with respect to modeling alignments, which contributes to previous work on understanding attention mechanisms (Ghader and Monz, 2017; Raganato and Tiedemann, 2018; Tang et al., 2018).

## 8 Conclusions

This paper addresses the task of jointly learning to produce translations and alignments with a single Transformer model. By using a multi-task objective along with providing full target sentence context to our alignment module, we are able to produce better alignments than previous approaches not relying on external alignment toolkits. We demonstrate that our framework can be extended to use external alignments from GIZA++ to achieve significantly better alignment results compared to GIZA++, while maintaining the same

translation performance.

Currently, our self-training based approach needs two training runs. To train our model in a single run, we would like to investigate a training method which alternates between alignment extraction and model training.

## Acknowledgments

We would like to thank the LILT team for releasing scripts and datasets for alignment evaluation. We are also grateful to Andrew Finch, Matthias Sperber, Barry Theobald and the anonymous reviewers for their helpful comments. Many thanks to Dorothea Peitz for helpful discussions about significance testing, Yi-Hsiu Liao for suggesting interesting extensions and the rest of Siri Machine Translation Team for their support.

## References

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 177–185, Brussels, Belgium.

Tamer Alkhouli and Hermann Ney. 2017. Biasing attention-based recurrent neural networks using external alignment information. In *Conference on Empirical Methods in Natural Language Processing*, pages 108–117, Copenhagen, Denmark.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, San Diego, CA, USA.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Conference on Statistical Machine Translation*, pages 272–303, Belgium, Brussels.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, and Frédéric Blain. 2017. Guiding neural machine translation decoding with external knowledge. In *Proceedings of the Second Conference on Machine Translation*, pages 157–168.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. In *Association for Machine Translation in the Americas*, pages 121–134, Austin, TX, USA.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, OH, USA.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *International Joint Conference on Natural Language Processing*, pages 30–39, Taipei, Taiwan.

Philipp Koehn, Amittai Axelrod, Ra Birch Mayne, Chris Callison-burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *International Workshop on Spoken Language Translation*, pages 68–75, Pittsburgh, PA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic.

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.

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. 2018. Target foresight based attention for neural machine translation. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1380–1390, New Orleans, LA, USA.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *International Conference on Computational Linguistics*, pages 3093–3102, Osaka, Japan.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*, New Orleans, LA, USA.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 2283–2288, Austin, TX, USA.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond - Volume 3*, pages 1–10, Edmonton, Canada.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA.

Jan-Thorsten Peter, Arne Nix Nix, and Hermann Ney. 2017. Generating alignments using target foresight in attention-based neural machine translation. In *Conference of the European Association for Machine Translation*, pages 27–36, Prague, Czech Republic.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Conference on Statistical Machine Translation*, pages 186–191, Belgium, Brussels.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 157–163, Valencia, Spain.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation. In *Conference on Statistical Machine Translation*, pages 26–35, Brussels, Belgium.

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)*, volume 1, pages 76–85.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 1–11, Long Beach, CA, USA.

David Vilar, Maja Popović, and Hermann Ney. 2006. AER: Do we need to "improve" our alignments? In *International Workshop on Spoken Language Translation*, pages 205–212, Kyoto, Japan.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *International Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding Interpretable Attention to Neural Translation Models Improves Word Alignment. *arXiv e-prints*.