Biasing Attention-Based Recurrent Neural Networks Using External Alignment Information

Tamer Alkhouli and Hermann Ney
Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
<surname>@i6.informatik.rwth-aachen.de

Abstract

This work explores extending attention-based neural models to include alignment information as input. We modify the attention component to have dependence on the current source position. The attention model is then used as a lexical model together with an additional alignment model to generate translation. The attention model is trained using external alignment information, and it is applied in decoding by performing beam search over the lexical and alignment hypotheses. The alignment model is used to score these alignment candidates. We demonstrate that the attention layer is capable of using the alignment information to improve over the baseline attention model that uses no such alignments. Our experiments are performed on two tasks: WMT 2016 English→Romanian and WMT 2017 German→English.

1 Introduction

Neural machine translation (NMT) has emerged recently as a successful end-to-end statistical machine translation approach. The best performing NMT systems use an attention mechanism that focuses the attention of the decoder on parts of the source sentence (Bahdanau et al., 2015). The attention component is computed as an intermediate part of the model, and is trained jointly with the rest of the model. The approach is appealing because (1) it is end-to-end, where the neural model is trained from scratch without assistance from other trained models, and (2) the attention component is trained jointly with the rest of the model, requiring no pre-computed alignments.

In this work, we raise the question whether the attention component is self-sufficient to attend to the source side, and if it can still benefit from explicit dependence on the alignment information. To this end, we modify the attention model to bias the attention layer towards the alignment information, and evaluate the model in a generative framework consisting of two steps: alignment prediction followed by lexical translation.

Two decades ago, (Vogel et al., 1996) applied hidden Markov models to machine translation. The idea was based on introducing word alignments as hidden variables, while using the first-order Markov assumption to simplify the dependencies of the alignment sequence. The approach decomposed the translation process using a lexical model and an alignment model. These models were simple tables enumerating all possible translation and alignment combinations. Nowadays, HMM is used with IBM models to generate word alignments, which are needed to train phrase-based systems.

Alkhouli et al. (2016) and Wang et al. (2017) apply the hidden Markov model decomposition using feedforward lexical and alignment neural network models. In this work, we are interested in using more expressive models. Namely, we leverage attention models as lexical models and use them with bidirectional recurrent alignment models. These recurrent models are able to encode unbounded source and target context in comparison to feedforward networks.

The attention-based translation model is conditioned on the full source sentence, but it has no explicit dependence on alignments as input. We propose to bias the attention mechanism using alignment information, while still allowing the model to compute attention weights dynamically. Conditioning the model on the alignment information as such makes it possible to combine with an alignment model in a generative story. We demonstrate that the attention model can benefit from such external alignment information on two WMT tasks: the 2016 English→Romanian task and the 2017 German→English task.

2 Related Work

Alignment-based neural models have explicit dependence on the alignment information either at the input or at the output of the network. They have been extensively and successfully applied in the literature on top of conventional phrase-based systems (Sundermeyer et al., 2014a; Tamura et al., 2014; Devlin et al., 2014). In this work, we focus on using the models directly to perform standalone neural machine translation.

Alignment-based neural models were proposed in (Alkhouli et al., 2016) to perform neural machine translation. They mainly used feedforward alignment and lexical models in decoding. In this work, we investigate recurrent models instead. We use a modified attention model as a lexical model and apply it together with a recurrent alignment neural model.

Deriving neural models for translation based on the HMM framework can also be found in (Yang et al., 2013; Yu et al., 2017). Alignment-based neural models were also applied to perform summarization and morphological inflection (Yu et al., 2016). The work used a monotonous alignment model, where training was done by marginalizing over the alignment hidden variables, which is computationally expensive. In this work, we use non-monotonous alignment models. In addition, we train using pre-computed Viterbi alignments which speeds up neural training. In (Yu et al., 2017), alignment-based neural models were used to model alignment and translation from the target to the source side (inverse direction), and a language model was included in addition. They showed results on a small translation task. In this work, we present results on translation tasks containing tens of millions of words. We do not include a language model in any of our systems.

There is plenty of work on modifying attention models to capture more complex dependencies. (Cohn et al., 2016) introduces structural biases from word-based alignment concepts like fertility and Markov conditioning. These are internal modifications that leave the model self-contained. Our modifications introduce alignments as external information to the model. (Arthur et al., 2016) include lexical probabilities to bias attention. (Chen et al., 2016; Mi et al., 2016) add an extra term dependent on the alignments to the training objective function to guide neural training. This is only applied during training but not during decoding. Our work modifies the attention component directly, and we can choose whether to apply the alignment bias during decoding or not. We show that using alignment bias during search alongside an alignment model improves translation.

3 Alignment-Based Translation

Given a source sentence $f_1^J = f_1...f_j...f_J$, a target sentence $e_1^I = e_1...e_i...e_I$, and an alignment sequence $b_1^I = b_1...b_i...b_I$, where $j = b_i$ is the source position aligned to the target position i, we model translation using an alignment model and a lexical model:

$$\begin{aligned} p(e_1^I|f_1^J) &= \sum_{b_1^I} p(e_1^I, b_1^I|f_1^J) \qquad (1) \\ &\approx \max_{b_1^I} \prod_{i=1}^{I} \underbrace{p(e_i|b_i, b_1^{i-1}, e_1^{i-1}, f_1^J)}_{\text{lexical model}} \cdot \\ &\qquad \underbrace{p(b_i|b_1^{i-1}, e_1^{i-1}, f_1^J)}_{\text{alignment model}} \end{aligned}$$

Both the lexical model and the alignment model have rich dependencies including the full source context f_1^J, the full alignment history b_1^{i-1}, and the full target history e_1^{i-1}. The lexical model has an extra dependence on the current source position b_i. First-order HMMs simplify the dependence on the alignment history and limit it to the predecessor alignment point b_{i-1}. This allows an efficient computation of the sum over the alignment sequence given in Eq. (1) using dynamic programming. In this work, we stick to the maximum approximation, and keep the full dependence on the alignment history b_1^{i-1}. We use recurrent neural networks to model the unbounded source, target and alignment context. Nevertheless, the models we describe can be simplified easily to drop the full dependence on the alignment history, in which case integrated training using the sum can be performed as suggested by Wang et al. (2017).

4 Attention-Based Translation Model

The standard attention-based translation model has three main components: The encoder, the decoder, and the attention component. The model

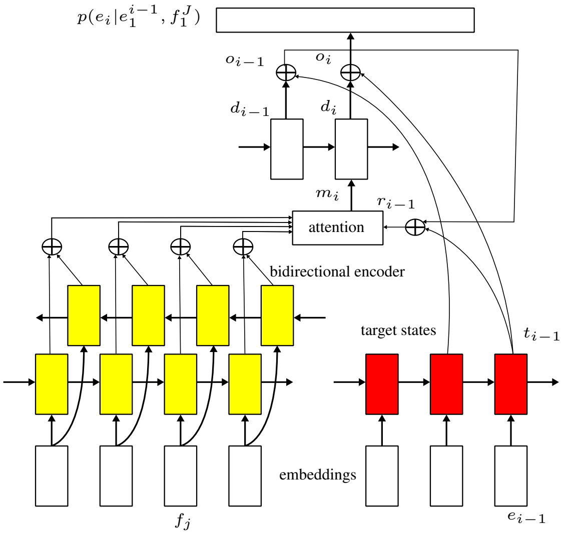

Figure 1: Attention model architecture.

architecture is illustrated in Fig. (1). We use long short-term memory (LSTM) recurrent layers throughout this work (Hochreiter and Schmidhuber, 1997; Gers et al., 2000, 2003). We include a bidirectional encoder where we sum the forward and backward source state representations:

$$\overrightarrow{h}_j = \text{LSTM}(\overrightarrow{h}_{j-1}, Ff_j)$$
$$\overleftarrow{h}_j = \text{LSTM}(\overleftarrow{h}_{j+1}, Ff_j)$$
$$h_j = Y\overleftarrow{h}_j + Z\overrightarrow{h}_j \quad (2)$$

where Y and Z are weight matrices, F is the source word embedding matrix, and $f_j \in \{0,1\}^{|V_f|\times 1}$ is the one-hot vector of the source word at position j. $|V_f|$ is the size of the source vocabulary. The parameterization of the recurrent layer is abstracted away using the LSTM notation for simplicity. We use an LSTM layer to represent the state of the target sequence:

$$t_{i-1} = \text{LSTM}(t_{i-2}, Ee_{i-1}) \quad (3)$$

where E is the target word embedding matrix, and $e_{i-1} \in \{0,1\}^{|V_e|\times 1}$ is the one-hot vector of the target word at position $i-1$. $|V_e|$ is the size of the target vocabulary. The attention weights are normalized using the softmax function according to the following equations:

$$\alpha_{ij} = \frac{\exp(s_{ij})}{\sum_{j=1}^{J}\exp(s_{ij})}$$
$$s_{ij} = v^T\tanh(Wh_j + Mr_{i-1} + a)$$
$$r_{i-1} = Ro_{i-1} + Lt_{i-1} \quad (4)$$
$$o_{i-1} = Ad_{i-1} + Bt_{i-2}$$
$$d_{i-1} = \text{LSTM}(d_{i-2}, m_{i-1}) \quad (5)$$
$$m_i = \sum_{j=1}^{J}\alpha_{ij}h_j$$

where α_{ij} denotes the normalized attention weights, s_{ij} denotes the unnormalized attention scores, r_{i-1} is the translation state computed using the decoder state at the previous step o_{i-1} and the target state t_{i-1} which in turn is computed using the target word e_{i-1}. The decoder state d_i is computed using an LSTM over the attended source positions m_i. v and a are vectors, and A, B, W, M, R, and L are weight matrices.

The final target word probability is computed as a softmax function of the decoder state $o_i \in \mathbb{R}^{|V_e|\times 1}$:

$$p(e_i = w|e_1^{i-1}, f_1^J) = \frac{\exp(o_{iw})}{\sum_{v=1}^{|V_e|}\exp(o_{iv})}$$

5 Alignment-Biased Attention

In order to use the attention model as an alignment-dependent lexical model, we introduce a dependence on the alignment information b_i. We modify the attention mechanism according to the following equation:

$$s_{ij} = v^T\tanh(Wh_j + Mr_{i-1} + a + \delta_{j,b_i}\,c) \quad (6)$$

where c is a vector, and δ_{j,b_i} is the Kronecker delta:

$$\delta_{j,b_i} = \begin{cases} 1, & \text{if} \quad j = b_i \\ 0, & \text{otherwise.} \end{cases}$$

We also experiment with a bias term that includes the aligned source state h_{b_i}:

$$s_{ij} = v^T\tanh(Wh_j + Mr_{i-1} + a + \delta_{j,b_i}Dh_{b_i}) \quad (7)$$

which we refer to as source alignment bias. D is an additional weight matrix. Note that the model will have full dependence on the alignment history due to Eq. (5) and Eq. (4) (cf. Fig. (1)). This dependency can be simplified by removing both the

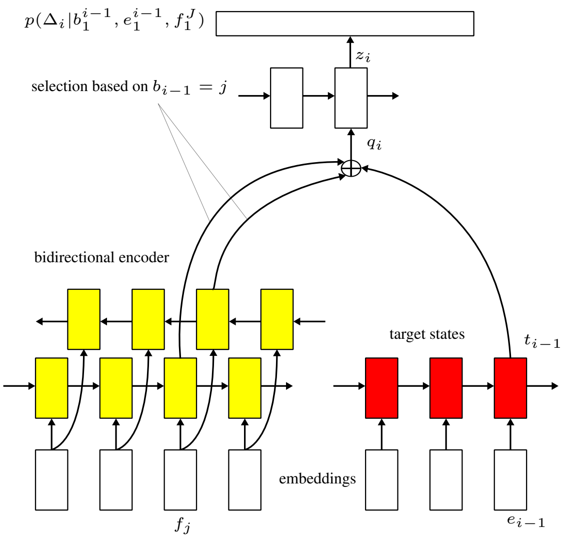

Figure 2: Bidirectional alignment model (BAM).

recurrency in Eq. (5), and the recurrent input o_{i-1} that feeds r_{i-1} in Eq. (4). In this work, however, we stick to the richer representation and keep the full dependence on the alignment history.

If the alignment information is pre-computed, e.g. through IBM/HMM training, using it as an alignment bias might risk that the original attention part will learn nothing and that it becomes completely dependent on the alignment information. To alleviate this problem, we include the alignment bias term during training for some batches and drop it for others. In our experiments, we randomly include the bias term for 50% of the training batches.

6 Recurrent Alignment Model

We use a recurrent alignment model to score alignments. The model architecture is shown in Fig. (2). Following (Alkhouli et al., 2016), the alignment model predicts the relative jump $\Delta_i = b_i - b_{i-1}$ from the previous source position b_{i-1} to the current source position b_i. This model has a bidirectional source encoder consisting of two recurrent layers (yellow), and a recurrent layer maintaining the target state (red). The most recent target state computed including word e_{i-1} is paired with the source states at position b_{i-1}, which is a hard alignment obtained externally and not computed by the model. We pair the source state h_j at position $j = b_{i-1}$ with the target state t_{i-1} at position $i-1$ to predict the jump Δ_i to the next source position b_i according to the following equations:

$$\begin{aligned} q_i &= Ut_{i-1} + h_{b_{i-1}} \\ z_i &= \text{LSTM}(z_{i-1}, q_i) \end{aligned} \tag{8}$$

where U is a weight matrix, q_i is the paired source and target states, and z_i is the decoder state used to predict the jump from b_{i-1} to b_i. $h_{b_{i-1}}$ and t_{i-1} are defined in Eq. (2), and Eq. (3), respectively. Removing the recurrency in Eq. (8) results in a first-order model over the alignment sequence.

7 Training

In this work, we train the attention and the alignment model separately. We obtain the alignments using IBM/HMM training. While this breaks up the simplicity of end-to-end training of attention models, we want to note that this is not central to the proposed approach. Integrated training using the sum instead of the maximum approximation in Eq. (1) can be performed using the Baum-Welch algorithm similar to (Yu et al., 2017; Wang et al., 2017), but the models need to give up the recurrency over the alignment information. Alternatively, the maximum approximation can be used to find the Viterbi alignments without changing the models, where training proceeds by alternating between aligning the training data and model estimation. In this work, however, we focus on the modeling aspect and leave integrated training to future work.

8 Alignment-Based Decoding

Similar to (Alkhouli et al., 2016), we combine the lexical and alignment neural models in a beam-based decoder. Since the models depend on the alignment information, we also have to hypothesize alignments during decoding. In training, we assume that each target position is aligned to exactly one source position. During decoding, we hypothesize all source positions for each target position. We assign the models separate weights and obtain the best translation as follows:

$$\begin{aligned} e_1^{\hat{I}} = \operatorname*{arg\,max}_{I, e_1^I} \Bigg\{ \frac{1}{I} \max_{b_1^I} \Bigg\{ \sum_{i=1}^{I} \lambda \log p(e_i|b_1^i, e_1^{i-1}, f_1^J) \\ + (1-\lambda) \log p(\Delta_i|b_1^{i-1}, e_1^{i-1}, f_1^J) \Bigg\} \Bigg\} \end{aligned} \tag{9}$$

where λ is the lexical model weight, which we tune on the development set using grid search.

	WMT 2016		WMT 2017	
	English	**Romanian**	**German**	**English**
Sentences	604K		3.55M	
Running Words	15.5M	15.8M	85M	86M
Vocabulary	92.3K	128.3K	671K	587K
Neural Network Vocabulary	56.1K	80.9K	188K	131K

Table 1: Corpora and NN statistics.

9 Experiments

9.1 Setup

This section presents experiments on two WMT shared translation tasks: the 2016 English→Romanian task[1] and the 2017 German→English task.[2] The corpora statistics are shown in Tab. (1). We use the full bilingual data of the English→Romanian task. For the German→English task, we choose the common crawl, news commentary and European parliament bilingual data. The data is filtered by removing sentences longer than 100 words. We also remove sentences where five or more consecutive source words are unaligned according to IBM1/HMM/IBM4 training. This is to remove noisy sentence pairs that are frequent in the common crawl corpus. We do not use any kind of synthetic or back-translated data in this work.

We reduce the vocabulary size by replacing singletons with the unknown token for both English and Romanian corpora in the English→Romanian task. Since we have more data in the German→English task, we replace words occurring less than 6 times in the German corpus and less than 4 times in the English corpus by the unknown token. The reduced vocabularies are what we refer to as the neural network vocabulary in Tab. (1). To handle the large output vocabularies, all lexical models use a class-factored output layer, with 1000 singleton classes dedicated to the most frequent words, and 1000 classes shared among the rest of the words. The classes are trained using a separate tool to optimize the maximum likelihood training criterion with the bigram assumption. The alignment model uses a small output layer of 201 nodes, determined by a maximum jump length of 100 (forward and backward). We train using stochastic gradient descent and halve the learning rate when the development perplexity increases.

We train feedforward models to compare to (Alkhouli et al., 2016). The models have two hidden layers, the first has 1000 nodes and the second has 500 nodes. We use a 9-word source window, and a 5-gram target history. 100 nodes are used for word embeddings. The bidirectional alignment models have 4 LSTM layers as shown in Fig. (2). We use 200-node source and target word embeddings and 200 nodes in each LSTM layer.

The attention models also use 200-node LSTM layers, and 200-node source and target embeddings. The internal dimension of the attention component is also set to 200 nodes, i.e. $v, a, c \in \mathbb{R}^{200\times 1}$.

Each model is trained on 4-12 CPU cores using the Intel MKL library, and takes about 2–4 days on average to converge.

We apply attention models with alignment bias and feedforward models in decoding using a decoder similar to that proposed in (Alkhouli et al., 2016). The decoder hypothesizes each source position for every target position being translated. Beam search is applied where the search nodes consist of both lexical and alignment hypotheses. When the attention model is applied without the alignment bias term, the decoder simplifies to hypothesizing lexical translations only. To speed up decoding of long sentences, we limit alignment hypotheses to the source positions $j \in \{i - 20, ..., i + 20\}$, where i is the current target position being translated. We use a beam size of 16 in all experiments. The alignments used during training are a result of IBM1/HMM/IBM4 training using GIZA++ (Och and Ney, 2003).

We use grid search to optimize the lexical model weights (cf. Eq. (9)). We find that the attention model receives a weight of 0.8, while the alignment model is assigned a weight of 0.2. We tune this on the development set of each task. We use 1000 sentence pairs of `newsdev2016` as the development set of the English→Romanian task, and `newstest2015` for tuning the German→English model weights.

[1] `http://www.statmt.org/wmt16/`

[2] `http://www.statmt.org/wmt17/`

These same datasets are used to halve the learning rate during model training.

All translation experiments are performed using an extension of the *Jane* toolkit (Vilar et al., 2010; Wuebker et al., 2012). The neural networks are trained using an extension of the *rwthlm* toolkit (Sundermeyer et al., 2014b). All results are measured in case-insensitive BLEU [%] (Papineni et al., 2002) using *mteval* from the *Moses* toolkit (Koehn et al., 2007). Case-insensitive TER [%] scores are computed with *TERCom* (Snover et al., 2006). Word classes are trained using an in-house tool (Botros et al., 2015) similar to *mkcls*.

9.2 Results

We compare our proposed system to three baseline systems on the WMT 2016 English→Romanian task and the WMT 2017 German→English task. The results are shown in Tab. (2). We set up a baseline system using a feedforward lexical model and a feedforward alignment model, to compare to the models used in (Alkhouli et al., 2016). This is shown in row 1. We first check the effect of using a recurrent alignment model (row 2) instead of the feedforward model. This brings an improvement of up to 1.6% BLEU. The attention baseline (row 3) performs much better in comparison, scoring up to 3.1% BLEU better than the feedforward system. This model has no alignment bias component. We note here that the German→English training data size is about 5.7 times more than that of the English→Romanian task, which can explain the small gap in performance between the systems in row 2 and row 3 on the German→English task, as the feeforward networks have large hidden layers of 1000 and 500 nodes, while the recurrent models use hidden layers of size 200.

We train an attention model by adding the alignment bias term in Eq. (6). We bias the attention model randomly during training for 50% of the training batches. During decoding, we include a bidirectional alignment model to score the alignment hypotheses (rows 4, 5). The combination of the alignment-biased attention model and the bidirectional alignment model (row 4) outperforms the standard attention model (row 3). This shows that the model learns to use the alignment information. We also compare to adding source alignment bias as given by Eq. (7) (row 5). We observe no difference to the case of constant alignment bias (row 4) on these tasks. Overall, we improve BLEU by 1.7% and 1.1% on the English→Romanian and the German→English task, respectively.

9.3 Alignment Model

In Tab. (3), we analyze the effect of the alignment model on the system. We observe that if the alignment model is dropped, the attention model is unable to score the alignments hypothesized during decoding on its own (row 4). If we drop the alignment model in decoding, we also have to exclude the alignment bias term when computing attention weights during decoding (row 3) (the bias term is still included in training). In this case, the translation degrades to the baseline performance.

9.4 Block out

In Tab. (3) we also investigate the effect of block out. On the English→Romanian task which has less training data in comparison to German→English, we observe that block out helps improve the system (row 2 vs. 5). This is because it avoids overfitting the alignment information, allowing the attention component to learn to attend on its own. This can be verified when comparing row 3 to row 6: When block out is used in training, and the attention model is used afterwards in decoding alone without an alignment model, it is able to perform close to the baseline attention performance if block out is used. Without using block out, the model fails to attend to the source side properly on its own.

9.5 Alignment Quality

We analyze the word alignment quality using 504 manually word-aligned German-English sentence pairs that were extracted from the Europarl corpus (Vilar et al., 2006). In Tab. (4), we compare the baseline attention system to the proposed alignment-based system. The alignments of the baseline attention system are generated by aligning each target word to the source position having the maximum attention weight. We observe that the baseline attention system has a high AER in comparison to the proposed system, which reduces AER from 44.9% to 29.7%. This corresponds to 1.1% BLEU improvement. It is worth noting that the high AER of the baseline system is likely because the model is not trained to align, and that the attention weights it produces are soft alignments. In comparison, our system uses an alignment model that explicitly learns to model alignments.

				WMT En→Ro newstest2016		WMT De→En newstest2017	
#	lexical model	alignment model	bias term	BLEU [%]	TER [%]	BLEU [%]	TER [%]
1	feedforward	feedforward	-	20.0	64.2	24.2	58.6
2	feedforward	bidirectional	-	21.6	62.7	25.5	57.6
3	attention	-	-	23.1	60.6	25.7	57.6
4	attention	bidirectional	$\delta_{j,bi}\, c$	24.8	58.1	26.8	55.6
5	attention	bidirectional	$\delta_{j,bi}\, D\, h_{b_i}$	24.8	58.1	26.8	55.5

Table 2: Translation results on the WMT 2016 English→Romanian task and the WMT 2017 German→English task.

					WMT En→Ro newstest2016		WMT De→En newstest2017	
#	lexical model	alignment model	decode w/ align bias	train w/ block out	BLEU [%]	TER [%]	BLEU [%]	TER [%]
1	attention baseline	-	-	-	23.1	60.6	25.7	57.6
2	+ alignment bias	bidirectional	yes	yes	24.8	58.1	26.8	55.6
3		-	no		23.1	60.6	25.7	59.4
4		-	yes		degenerate		degenerate	
5		bidirectional	yes	no	23.7	59.2	26.7	55.8
6		-	no		degenerate		degenerate	

Table 3: The effect of using the alignment model in decoding and block out in training . The alignment bias term used here is $\delta_{j,bi}\, c$. Rows 1 and 2 are the same as rows 3 and 4 in Tab. (2). Block out means including the alignment bias term for 50% of the training batches.

	newstest2017 BLEU [%]	Europarl AER [%]
attention baseline	25.7	44.9
proposed system	26.8	29.7

Table 4: A comparison between the WMT German→English proposed system and the baseline attention system in terms of the alignment error rate (AER). The attention baseline and the proposed system are the same ones shown in Tab. (2), rows 3 and 4, respectively.

To illustrate what happens when we include the source alignment bias term, we take a sample from the translation hypotheses of the German→English system in Tab. (2, row 5), and compare it to the output of the standard attention model Tab. (2, row 3). The sample is chosen from the development set `newstest2015`. The German sentence "*diese schreckliche Erfahrung wird uns immer verfolgen .*" has the reference translation " *this horrible experience will stay with us .*"

In Fig. (3), we illustrate the best translation hypothesis and the corresponding attention weights produced by the standard attention model. Fig. (4) shows the same thing for the attention model using source alignment bias. We observe that the latter is able to generate a good translation while being able to attend to the source sentence in a proper order. On the other hand, the standard attention model has a problem in the first half of the hypothesis, where it attends to the second half of the source sentence instead. It ends up confusing the object and the subject. A more acceptable, though inaccurate, translation of 'verfolgen' under such reordering would be 'followed by', but the system fails to generate this translation.

Fig. (5) shows the curve of tuning the lexical model weight. We observe that the weight is robust against small changes. The best results in terms of BLEU are achieved when $\lambda = 0.8$.

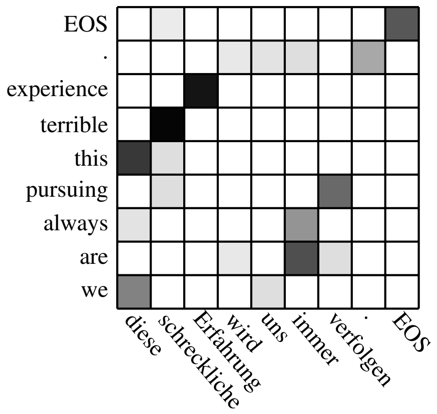

Figure 3: A translation example produced by the standard attention system in Tab. (2), row 3. EOS denotes the sentence end symbol. The shading degree corresponds to the attention weight.

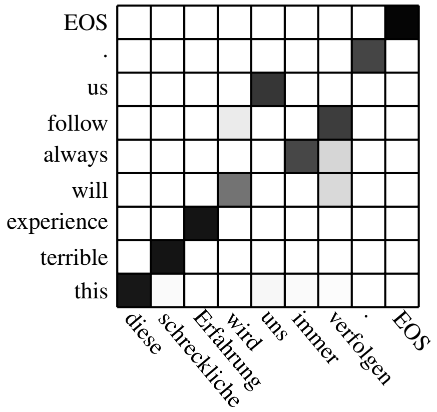

Figure 4: A translation example produced by our best system using source alignment bias, given in Tab. (2), row 5. EOS denotes the sentence end symbol. The shading degree corresponds to the attention weight.

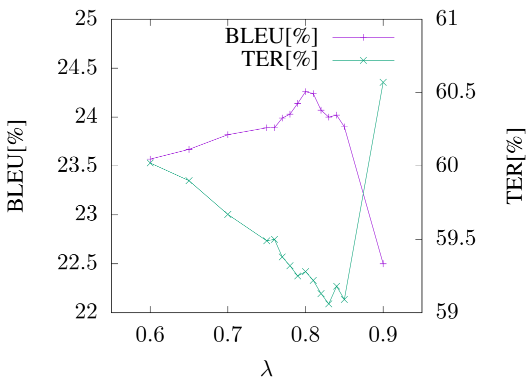

Figure 5: Grid search tuning of the lexical weight of the system in Tab. (2, row 4). The results are computed on the development set of the English→Romanian task.

10 Conclusion

We presented a modification of the attention model to bias it using external alignment information. We also presented a bidirectional recurrent neural network alignment model to be used alongside the proposed attention model. We used the two models in a generative scheme of alignment generation followed by lexical translation. We demonstrated improvements over the standard attention model on two WMT tasks. We provided evidence that enabling the alignment bias term for all training samples makes the attention mechanism overfit the alignments on non-large datasets. To remedy this, we proposed to apply the alignment bias on half of the training samples, which yielded our best system.

While this work depends on pre-computed alignments to train the attention and alignment models, this is not central to our approach. In future work, we plan to perform integrated training by alternating between alignment generation and model estimation. Alignment generation can be performed using forced alignment where beam search is performed over the alignment positions, while fixing the lexical translations to the reference translation. This can eliminate the need for pre-computing alignments using ad hoc methods like IBM1/ HMM/IBM4 training.

Acknowledgements

The work reported in this paper results from two projects, SEQCLAS and QT21. SEQCLAS has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement n^{o} 694537. QT21 has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n^{o} 645452. The work reflects only the authors' views and neither the European Commission nor the European Research Council Executive Agency are responsible for any use that may be made of the information it contains.

Tamer Alkhouli was partly funded by the 2016 Google PhD Fellowship for North America, Europe and the Middle East.

The authors would like to thank Kazuki Irie for contributing to the attention layer implementation.

References

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.

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, Austin, Texas.

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, Calefornia, USA.

Rami Botros, Kazuki Irie, Martin Sundermeyer, and Hermann Ney. 2015. On efficient training of word classes and their application to recurrent neural network language models. In *Interspeech*, pages 1443–1447, Dresden, Germany.

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

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.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *52nd Annual Meeting of the Association for Computational Linguistics*, pages 1370–1380, Baltimore, MD, USA.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471.

Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. 2003. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantine, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. pages 177–180, Prague, Czech Republic.

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.

Franz J. 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 *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014a. Translation Modeling with Bidirectional Recurrent Neural Networks. In *Conference on Empirical Methods on Natural Language Processing*, pages 14–25, Doha, Qatar.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2014b. rwthlm - the RWTH Aachen university neural network language modeling toolkit. In *Interspeech*, pages 2093–2097, Singapore.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2014. Recurrent neural networks for word alignment model. In *52nd Annual Meeting of the Association for Computational Linguistics*, pages 1470–1480, Baltimore, MD, USA.

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

David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. 2010. Jane: Open source hierarchical translation, extended with reordering and lexicon models. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 262–270, Uppsala, Sweden.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *Proceedings of the 16th conference on Computational linguistics*, volume 2, pages 836–841, Copenhagen, Denmark.

Weiyue Wang, Tamer Alkhouli, Derui Zhu, and Hermann Ney. 2017. Hybrid neural network alignment and lexicon model in direct hmm for statistical machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada.

Joern Wuebker, Matthias Huck, Stephan Peitz, Malte Nuhn, Markus Freitag, Jan-Thorsten Peter, Saab Mansour, and Hermann Ney. 2012. Jane 2: Open source phrase-based and hierarchical statistical machine translation. In *International Conference on Computational Linguistics*, pages 483–491, Mumbai, India.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *51st Annual Meeting of the Association for Computational Linguistics*, pages 166–175, Sofia, Bulgaria.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomás Kociský. 2017. The neural noisy channel. In *Proceedings of the International Conference on Learning Representations*, volume abs/1611.02554.

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.