Incorporating Structural Alignment Biases into an Attentional Neural Translation Model

Trevor Cohn and **Cong Duy Vu Hoang** and **Ekaterina Vymolova**
University of Melbourne
Melbourne, VIC, Australia
tcohn@unimelb.edu.au and {vhoang2,evylomova}@student.unimelb.edu.au

Kaisheng Yao
Microsoft Research
Redmond, WA, USA
kaisheng.YAO@microsoft.com

Chris Dyer
Carnegie Mellon University
Pittsburgh, PA, USA
cdyer@cmu.edu

Gholamreza Haffari
Monash University
Clayton, VIC, Australia
gholamreza.haffari@monash.edu

Abstract

Neural encoder-decoder models of machine translation have achieved impressive results, rivalling traditional translation models. However their modelling formulation is overly simplistic, and omits several key inductive biases built into traditional models. In this paper we extend the attentional neural translation model to include structural biases from word based alignment models, including positional bias, Markov conditioning, fertility and agreement over translation directions. We show improvements over a baseline attentional model and standard phrase-based model over several language pairs, evaluating on difficult languages in a low resource setting.

1 Introduction

Recently, models of end-to-end machine translation based on neural network classification have been shown to produce excellent translations, rivalling or in some cases surpassing traditional statistical machine translation systems (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). This is despite the neural approaches using an overall simpler model, with fewer assumptions about the learning and prediction problem.

Broadly, neural approaches are based around the notion of an *encoder-decoder* (Sutskever et al., 2014), in which the source language is *encoded* into a distributed representation, followed by a *decoding* step which generates the target translation. We focus on the *attentional model* of translation (Bahdanau et al., 2015) which uses a dynamic representation of the source sentence while allowing the decoder to *attend* to different parts of the source as it generates the target sentence. The attentional model raises intriguing opportunities, given the correspondence between the notions of attention and alignment in traditional word-based machine translation models (Brown et al., 1993).

In this paper we map modelling biases from word based translation models into the attentional model, such that known linguistic elements of translation can be better captured. We incorporate *absolute positional bias* whereby word order tends to be similar between the source sentence and its translation (e.g., IBM Model 2 and (Dyer et al., 2013)), *fertility* whereby each instance of a source word type tends to be translated into a consistent number of target tokens (e.g., IBM Models 3, 4, 5), *relative position bias* whereby prior preferences for monotonic alignments/attention can be encouraged (e.g., IBM Model 4, 5 and HMM-based Alignment (Vogel et al., 1996)), and *alignment consistency* whereby the attention in *both* translation directions are encouraged to agree (e.g. symmetrisation heuristics (Och and Ney, 2003) or joint modelling (Liang et al., 2006; Ganchev et al., 2008)).

We provide an empirical analysis of incorporating the above structural biases into the attentional model, considering low resource translation scenario over four language-pairs. Our results demonstrate consistent improvements over vanilla encoder-

Proceedings of NAACL-HLT 2016, pages 876–885,
San Diego, California, June 12-17, 2016. ©2016 Association for Computational Linguistics

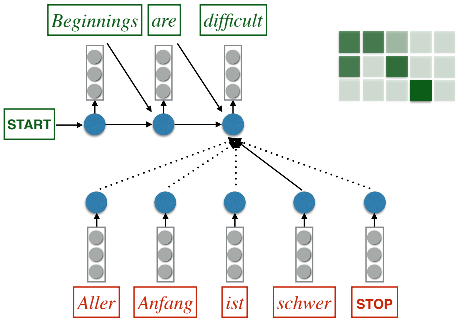

Figure 1: Attentional model of translation (Bahdanau et al., 2015). The encoder is shown below the decoder, and the edges connecting the two corresponding to the attention mechanism. Heavy edges denote a higher attention weight, and these values are also displayed in matrix form, with one row for each target word.

decoder and attentional model in terms of the perplexity and BLEU score, e.g. up to 3.5 BLEU points when re-ranking the candidate translations generated by a state-of-the-art phrase based model.

2 The attentional model of translation

We start by reviewing the attentional model of translation (Bahdanau et al., 2015), as illustrated in Fig. 1, before presenting our extensions in §3.

Encoder The encoding of the source sentence is formulated using a pair of RNNs (denoted *bi-RNN*) one operating left-to-right over the input sequence and another operating right-to-left,

$$\begin{aligned}\boldsymbol{h}_i^{\rightarrow} &= \text{RNN}(\boldsymbol{h}_{i-1}^{\rightarrow}, \boldsymbol{r}^{(\text{s})}_{s_i})\\ \boldsymbol{h}_i^{\leftarrow} &= \text{RNN}(\boldsymbol{h}_{i+1}^{\rightarrow}, \boldsymbol{r}^{(\text{s})}_{s_i})\end{aligned}$$

where $\boldsymbol{h}_i^{\rightarrow}$ and $\boldsymbol{h}_i^{\leftarrow}$ are the RNN hidden states. The left-to-right RNN function is defined as

$$\boldsymbol{h}_i^{\rightarrow} = \tanh\left(\boldsymbol{W}_{si}^{\rightarrow}\boldsymbol{r}^{(\text{s})}_{s_i} + \boldsymbol{W}_{sh}^{\rightarrow}\boldsymbol{h}_{i-1}^{\rightarrow} + \boldsymbol{b}_s^{\rightarrow}\right) \quad (1)$$

where $\boldsymbol{h}_0^{\rightarrow} \in \mathbb{R}^H$ is a learned parameter vector, as are $\mathbf{R}^{(\text{s})} \in \mathbb{R}^{V_S \times E}$, $\boldsymbol{W}_{si}^{\rightarrow} \in \mathbb{R}^{H \times E}$, $\boldsymbol{W}_{sh}^{\rightarrow} \in \mathbb{R}^{H \times H}$ and $\boldsymbol{b}_s^{\rightarrow} \in \mathbb{R}^H$, with H the number of hidden units, V_S the size of the source vocabulary and E the word embedding dimensionality.[1] Each source word is then represented as a pair of hidden states, one from each RNN, $\boldsymbol{e}_i = \begin{bmatrix}\boldsymbol{h}_i^{\rightarrow}\\ \boldsymbol{h}_i^{\leftarrow}\end{bmatrix}$. This encodes not only the word but also its left and right context, which can provide important evidence for its translation.

A crucial question is how this dynamic sized matrix $\mathbf{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_I] \in \mathbb{R}^{I \times H}$ can be used in the decoder to generate the target sentence. As with Sutskever's encoder-decoder, the target sentence is created left-to-right using an RNN, while the encoded source is used to bias the process as an auxiliary input. The mechanism for this bias is by attentional vectors, i.e. vectors of scores over each source sentence location, which are used to aggregate the dynamic source encoding into a fixed length vector.

Decoder The decoder operates as a standard RNN over the translation $\boldsymbol{t}$, formulated as follows

$$\boldsymbol{g}_j = \tanh\left(\mathbf{W}^{(\text{th})}\boldsymbol{g}_{j-1} + \mathbf{W}^{(\text{ti})}\boldsymbol{r}^{(\text{t})}_{t_{j-1}} + \mathbf{W}^{(\text{ta})}\boldsymbol{c}_j\right) \quad (2)$$

$$\boldsymbol{u}_j = \tanh\left(\boldsymbol{g}_j + \mathbf{W}^{(\text{uc})}\boldsymbol{c}_j + \mathbf{W}^{(\text{ui})}\boldsymbol{r}^{(\text{t})}_{t_{j-1}}\right) \quad (3)$$

$$t_j \sim \text{softmax}\left(\mathbf{W}^{(\text{ou})}\boldsymbol{u}_j + \boldsymbol{b}^{(\text{to})}\right) \quad (4)$$

where the decoder RNN is defined analogously to Eq 1 but with an additional input, the source attention component $\boldsymbol{c}_j \in \mathbb{R}^{2H}$ and weighting matrix $\mathbf{W}^{(\text{ta})} \in \mathbb{R}^{H \times 2H}$. The hidden state of the recurrence is then passed through a single hidden layer[2] (Eq 3) in combination with the source attention and target word using weighting matrices $\mathbf{W}^{(\text{uc})} \in \mathbb{R}^{H \times 2H}$ and $\mathbf{W}^{(\text{ui})} \in \mathbb{R}^{H \times E}$. In Eq 4 this vector is transformed to be target vocabulary sized, using weight matrix $\mathbf{W}^{(\text{ou})} \in \mathbb{R}^{V_T \times H}$ and bias $\boldsymbol{b}^{(\text{to})} \in \mathbb{R}^{V_T}$, after which a softmax is taken, and the resulting normalised vector used as the parameters of a Categorical distribution in generating the next target word.

The presentation above assumes a simple RNN is used to define the recurrence over hidden states, however we can easily use alternative formulations of recurrent networks including multiple-layer RNNs, gated recurrent units (GRU; Cho et al. (2014)), or long short-term memory (LSTM; Hochreiter and Schmidhuber (1997)) units. These more advanced methods allow for more efficient learning of more complex concepts, particularly

[1] Similarly, $\boldsymbol{h}_0^{\leftarrow} \in \mathbb{R}^H, \boldsymbol{W}_{si}^{\leftarrow} \in \mathbb{R}^{H \times E}, \boldsymbol{W}_{sh}^{\leftarrow} \in \mathbb{R}^{H \times H}, \boldsymbol{b}_s^{\leftarrow} \in \mathbb{R}^H$ are the parameters of the right-to-left RNN. Note that we use a long short term memory unit (Hochreiter and Schmidhuber, 1997) in place of the RNN, shown here for simplicity of exposition.

[2] In Bahdanau et al. (2015) they use a max-out layer for this final step, however we found this to be a needless complication, and instead use a standard hidden layer with tanh activation.

long distance effects. Empirically we found LSTMs to be the best performing, and therefore use these units herein.

The last key detail is the attentional component $\boldsymbol{c}_j$ in Eqs 2 and 3, which is defined as follows

$$\begin{aligned} f_{ji} &= \boldsymbol{v}^\top \tanh\left(\mathbf{W}^{(\text{ae})}\boldsymbol{e}_i + \mathbf{W}^{(\text{ah})}\boldsymbol{g}_{j-1}\right) \qquad (5) \\ \boldsymbol{\alpha}_j &= \operatorname{softmax}(\boldsymbol{f}_j) \\ \boldsymbol{c}_j &= \sum_i \alpha_{ji}\boldsymbol{e}_i \end{aligned}$$

with the scalars f_{ji} denoting the compatibility between the target hidden state $\boldsymbol{g}_{j-1}$ and the source encoding $\boldsymbol{e}_i$. This is defined as a neural network with one hidden layer of size A and a single output, parameterised by $\mathbf{W}^{(\text{ae})} \in \mathbb{R}^{A\times 2H}$, $\mathbf{W}^{(\text{ah})} \in \mathbb{R}^{A\times H}$ and $\boldsymbol{v} \in \mathbb{R}^A$. The softmax then normalises the scalar compatibility values such that for a given target word j, the values of α_j can be interpreted as alignment probabilities to each source location. Finally, these alignments are used to to reweight the source components E to produce a fixed length context representation.

Training of this model is done by minimising the cross-entropy of the target sentence, measured word-by-word as for a language model. We use standard stochastic gradient optimisation using the back-propagation technique for computation of partial derivatives according to the chain rule.

3 Incorporating Structural Biases

The attentional model, as described above, provides a powerful and elegant model of translation in which alignments between source and target words are learned through the implicit conditioning context afforded by the attention mechanism. Despite its elegance, the attentional model omits several key components of a traditional alignment models such as the IBM models (Brown et al., 1993) and Vogel's hidden Markov Model (Vogel et al., 1996) as implemented in the GIZA++ toolkit (Och and Ney, 2003). Combining the strengths of this highly successful body of research into a neural model of machine translation holds potential to further improve modelling accuracy of neural techniques. Below we outline methods for incorporating these factors as structural biases into the attentional model.

3.1 Position bias

First we consider position bias, based on the observation that a word at a given relative position in the source tends to align to a word at a similar relative position in the target, $\frac{i}{I} \approx \frac{j}{J}$ (Dyer et al., 2013). Related, the IBM model 2 learns discrete mappings between positions i and j conditioned on sentence lengths I and J.

We include a position bias through redefining the pre-normalised attention scalars f_{ji} in Eq 5 as:

$$\begin{aligned} f_{ji} = \boldsymbol{v}^\top \tanh\big(&\mathbf{W}^{(\text{ae})}\boldsymbol{e}_i + \mathbf{W}^{(\text{ah})}\boldsymbol{g}_{j-1} + \\ &\mathbf{W}^{(\text{ap})}\psi(j,i,I)\big) \qquad (6) \end{aligned}$$

where the new component in the input is a simple feature function of the positions in the source and target sentences and the source length,

$$\psi(j,i,I) = \Big[\log(1+j), \log(1+i), \log(1+I)\Big]^\top$$

and $\mathbf{W}^{(\text{ap})} \in \mathbb{R}^{A\times 3}$. We exclude the target length J as this is unknown during decoding, as a partial translation can have several (infinite) different lengths. The use of the $\log(1+\cdot)$ function is to avoid numerical instabilities from widely varying sentence lengths. The non-linearity in Eq 6 allows for complex functions of these inputs to be learned, such as relative positions and approximate distance from the diagonal, as well as their interactions with the other inputs (e.g., to learn that some words are exceptional cases where a diagonal bias should not apply).

3.2 Markov condition

The HMM model of translation (Vogel et al., 1996) is based on a Markov condition over alignment random variables, to allow the model to learn local effects such as when $i \leftarrow j$ is aligned then it is likely that $i+1 \leftarrow j+1$ or $i \leftarrow j+1$. These correspond to local diagonal alignments or one-to-many alignments, respectively. In general, there are many correlations between the alignments of a word and the alignments of the preceding word.

Markov conditioning can also be incorporated in a similar manner to positional bias, by augmenting the attentional input from Eqs 5 and 6 to include:

$$f_{ji} = \boldsymbol{v}^\top \tanh\left(\ldots + \mathbf{W}^{(\text{am})}\xi_1(\boldsymbol{\alpha}_{j-1}; i)\right) \qquad (7)$$

where $\ldots$ abbreviates the $\boldsymbol{e}_i$, $\boldsymbol{g}_{j-1}$ and ψ components from Eq 6, and $\xi_1(\boldsymbol{\alpha}_{j-1})$ provides a fixed dimensional representation of the attention state for the preceding word. It is not immediately obvious how to incorporate the previous attention vector as $\boldsymbol{\alpha}$ is dynamically sized to match the source sentence length, thus using it directly would not generalise over sentences of different lengths. For this reason, we make a simplification by just considering local moves offset by $\pm k$ positions, that is,

$$\xi_1(\boldsymbol{\alpha}_{j-1}; i) = \left[\alpha_{j-1,i-k}, .., \alpha_{j-1,i}, .., \alpha_{j-1,i+k}\right]^\top$$

with $\mathbf{W}^{(\mathrm{am})} \in \mathbb{R}^{A\times(2k+1)}$. Our approach is likely to capture the most important alignments patterns forming the backbone of the alignment HMM, namely monotone, 1-to-many, and local inversions.

3.3 Fertility

Fertility is the propensity for a word to be translated as a consistent number of words in the other language, e.g., *Iseseisvusdeklaratsioon* (Et) translates as 3-4 words in English, namely *(the) Declaration of Independence*. Fertility is a central component in the IBM models 3–5 (Brown et al., 1993). Incorporating fertility into the attentional model is a little more involved, and we present two techniques for doing so.

Local fertility First we consider a feature-based technique, which includes the following features

$$\xi_2(\boldsymbol{\alpha}_{<j}; i) = \left[\sum_{j'<j} \alpha_{j',i-k}, .., \sum_{j'<j} \alpha_{j',i}, .., \sum_{j'<j} \alpha_{j',i+1}\right]^\top$$

and the corresponding feature weights, i.e., $\mathbf{W}^{(\mathrm{af})} \in \mathbb{R}^{A\times(2k+1)}$. These sums represent the total alignment score for the surrounding source words, similar to fertility in a traditional latent variable model, which is the sum over binary alignment random variables. A word which already has several alignments can be excluded from participating in more alignments, thus combating the garbage collection problem. Conversely words that tend to need high fertility can be learned through the interactions between these features and the word and context embeddings in Eq 7.

Global fertility A second, more explicit, technique for incorporating fertility is to include this as a modelling constraint. Initially we considered a soft constraint based on the approach in (Xu et al., 2015), where an image captioning model was biased to attend to every pixel in the image exactly once. In our setting, the same idea can be applied through adding a regularisation term to the training objective of the form $\sum_i \left(1 - \sum_j \alpha_{j,i}\right)^2$. However this method is overly restrictive: enforcing that every word is used exactly once is not appropriate in translation where some words are likely to be dropped (e.g., determiners and other function words), while others might need to be translated several times to produce a phrase in the target language.[3] For this reason we develop an alternative method, based around a contextual fertility model, $p(f_i|\boldsymbol{s}, i) = \mathcal{N}\left(\mu(e_i), \sigma^2(e_i)\right)$ which scores the fertility of source word i, defined as $f_i = \sum_j \alpha_{j,i}$, using a normal distribution[4] parameterised by μ and σ^2, both positive scalar valued non-linear functions of the source word encoding e_i. This is incorporated into the training objective as an additional additive term, $\sum_i \log p(f_i|\boldsymbol{s}, i)$, for each training sentence.

This formulation allows for greater consistency in translation, through e.g., learning which words tend to be omitted from translation, or translate as several words. Compared to the fertility model in IBM 3–5 (Brown et al., 1993), ours uses many fewer parameters through working over vector embeddings, and moreover, the BiRNN encoding of the source means that we learn context-dependent fertilities, which can be useful for dealing with fixed syntactic patterns or multi-word expressions.

3.4 Bilingual Symmetry

So far we have considered a conditional model of the target given the source, modelling $p(\boldsymbol{t}|\boldsymbol{s})$. However it is well established for latent variable translation models that the alignments improve if $p(\boldsymbol{s}|\boldsymbol{t})$ is

[3] Modern decoders (Koehn et al., 2003) often impose the restriction of each word being translated exactly once, however this is tempered by their use of phrases as translation units rather than words, which allow for higher fertility within phrases.

[4] The normal distribution is deficient, as it has support for all scalar values, despite f_i being bounded above and below ($0 \leq f_i \leq J$). This could be corrected by using a truncated normal, or various other choices of distribution.

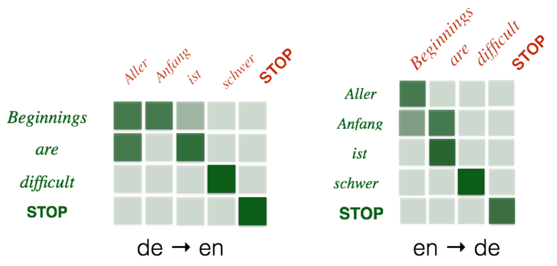

Figure 2: Symmetric training with trace bonus, computed as matrix multiplication, $-\operatorname{tr}(\boldsymbol{\alpha}^{s\leftarrow t}\boldsymbol{\alpha}^{s\rightarrow t\,\top})$. Dark shading indicates higher values.

also modelled and the inferences of both directional models are combined – evidenced by the symmetrisation heuristics used in most decoders (Koehn et al., 2005), and also by explicit joint agreement training objectives (Liang et al., 2006; Ganchev et al., 2008). The rationale is that both models make somewhat independent errors, so an ensemble stands to gain from variance reduction.

We propose a method for joint training of two directional models as pictured in Figure 2. Training twinned models involves optimising $\mathcal{L} = -\log p(\boldsymbol{t}|\boldsymbol{s}) - \log p(\boldsymbol{s}|\boldsymbol{t}) + \gamma B$ where, as before, we consider only a single sentence pair, for simplicity of notation. This corresponds to a pseudo-likelihood objective, with the B linking the two models.[5] The B component considers the alignment (attention) matrices, $\boldsymbol{\alpha}^{s\rightarrow t} \in \mathbb{R}^{J\times I}$ and $\boldsymbol{\alpha}^{t\leftarrow s} \in \mathbb{R}^{I\times J}$, and attempts to make these close to one another for both translation directions (see Fig. 2). To achieve this, we use a 'trace bonus', inspired by (Levinboim et al., 2015), formulated as

$$B = -\operatorname{tr}(\boldsymbol{\alpha}^{s\leftarrow t\,\top}\boldsymbol{\alpha}^{s\rightarrow t}) = \sum_j \sum_i \alpha^{s\leftarrow t}_{i,j}\,\alpha^{s\rightarrow t}_{j,i}\,.$$

As the alignment cells are normalised using the $\mathrm{softmax}$ and thus take values in [0,1], the trace term is bounded above by $\min(I, J)$ which occurs when the two alignment matrices are transposes of each other, representing perfect one-to-one alignments in both directions

lang-pair	# tokens (K)		# types (K)	
Zh-En	422	454	3.44	3.12
Ru-En	1639	1809	145	65
Et-En	1411	1857	90	25
Ro-En	1782	1806	39	24

Table 1: Statistics of the training sets, showing in each cell the count for the source language (left) and target language (right).

4 Experiments

Datasets. We conducted our experiments with four language pairs, translating between English ↔ Romanian, Estonian, Russian and Chinese. These languages were chosen to represent a range of translation difficulties, including languages with significant morphological complexity (Estonian, Russian). We focus on a (simulated) low resource setting, where only a limited amount of training data is available. This serves to demonstrate the robustness and generalisation of our model on sparse data – something that has not yet been established for neural models with millions of parameters with vast potential for over-fitting.

Table 1 shows the statistics of the training sets.[6] For Chinese-English, the data comes from the BTEC corpus, where the number of training sentence pairs is 44,016. We used 'devset1_2' and 'devset_3' as the development and test sets, respectively, and in both cases used only the first reference for evaluation. For Romanian and Estonian, the data come from the Europarl corpus (Koehn, 2005), where we used 100K sentence pairs for training, and 3K for development and 2K for testing.[7] The Russian-English data was taken from a web derived corpus (Antonova and Misyurev, 2011). The dataset is split into three parts using the same technique as for the Europarl sets. During the preprocessing stage we lower-cased and tokenized the data, and excluded sentences longer than 30 words. For the Europarl

[5] We could share some parameters, e.g., the word embedding matrices, however we found this didn't make much difference versus using disjoint parameter sets. We set $\gamma = 1$ herein.

[6] For all datasets words were thresholded for training frequency ≥ 5, with uncommon training and unseen testing words replaced by an $\langle$unk$\rangle$ symbol.

[7] The first 100K sentence pairs were used for training, while the development and test were drawn from the last 100K sentence pairs, taking the first 2K for testing and the last 3K for development.

data, we also removed sentences containing headings and other meeting formalities.[8]

Models and Baselines. We have implemented our neural translation model with linguistic features in C++ using the CNN library.[9] We compared our proposed model against our implementations of the attentional model (Bahdanau et al., 2015) and encoder-decoder architecture (Sutskever et al., 2014). As the baseline, we used a state-of-the-art phrase-based statistical machine translation model built using Moses (Koehn et al., 2007) with the standard features: relative-frequency and lexical translation model probabilities in both directions; distortion model; language model and word count. We used KenLM (Heafield, 2011) to create 3-gram language models with Kneser-Ney smoothing on the target side of the bilingual training corpora.

Evaluation Measures. Following previous work (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Neubig et al., 2015), we evaluated all neural models using test set perplexities and translation results, as well as in an additional re-ranking setting, using BLEU (Papineni et al., 2002) measure. We applied bootstrap resampling (Koehn, 2004) to measure statistical significance, $p < 0.05$, of our models compared to a baseline. For re-ranking, we generated 100-best translations using the baseline phrase-based model, to which we added log probability features from our neural models alongside all the features of the underlying phrase-based model. We trained the re-ranking models using MERT (Och, 2003) on development sets with 100-best translations.

4.1 Analysis of Alignment Biases

We start by investigating the effect of various linguistic constraints, described in Section 3, on the attentional model. Table 2 presents the perplexity of trained models for Chinese→English translation. For comparison, we report the results of an encoder-decoder-based neural translation model (Sutskever et al., 2014) as the baseline. All other results are for the attentional model with a single-layer LSTM as encoder and two-layer LSTM as decoder, using 512 embedding, 512 hidden, and 256 alignment dimensions. For each model, we also report the number of its parameters. Models are trained end-to-end using stochastic gradient descent (SGD), allowing up to 20 epochs. We use a held-out development set for regularisation by early stopping, which terminated the training after 5-10 epochs for most cases.

configuration	**test**	**#param (M)**
Sutskever encdec	5.35	8.7
Attentional	4.77	15.0
+align	4.56	15.0
+align+glofer	5.20	15.5
+align+glofer-pre	4.31	15.5
+align+sym	4.44	30.1
+align+sym+glofer-pre	4.43	31.2

Table 2: Perplexity results for attentional model variants evaluated on BTEC zh→en, and number of model parameters (in millions).

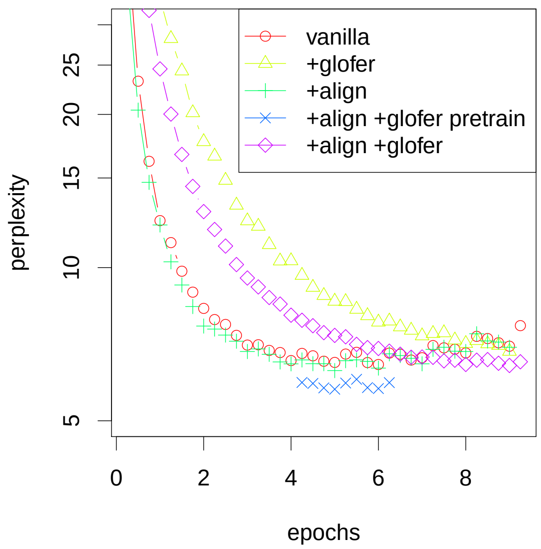

Figure 3: Perplexity with training epochs on ro-en translation, comparing several model variants.

As expected, the vanilla attentional model greatly improves over encoder-decoder (perplexity of 4.77 vs. 5.35), clearly making good use of the additional context. Adding the combined positional bias, local fertility, and Markov structure (denoted by +align) further decreases the perplexity to 4.56. Adding the global fertility (+glofer) is detrimental, however, increases perplexity to 5.20. Interestingly, global fertility helps to reduce the perplexity (to 4.31) when used with the pre-training setting (+align+glofer-

[8] E.g., *(The sitting was closed at 10.20pm)*.

[9] `https://github.com/clab/cnn/`

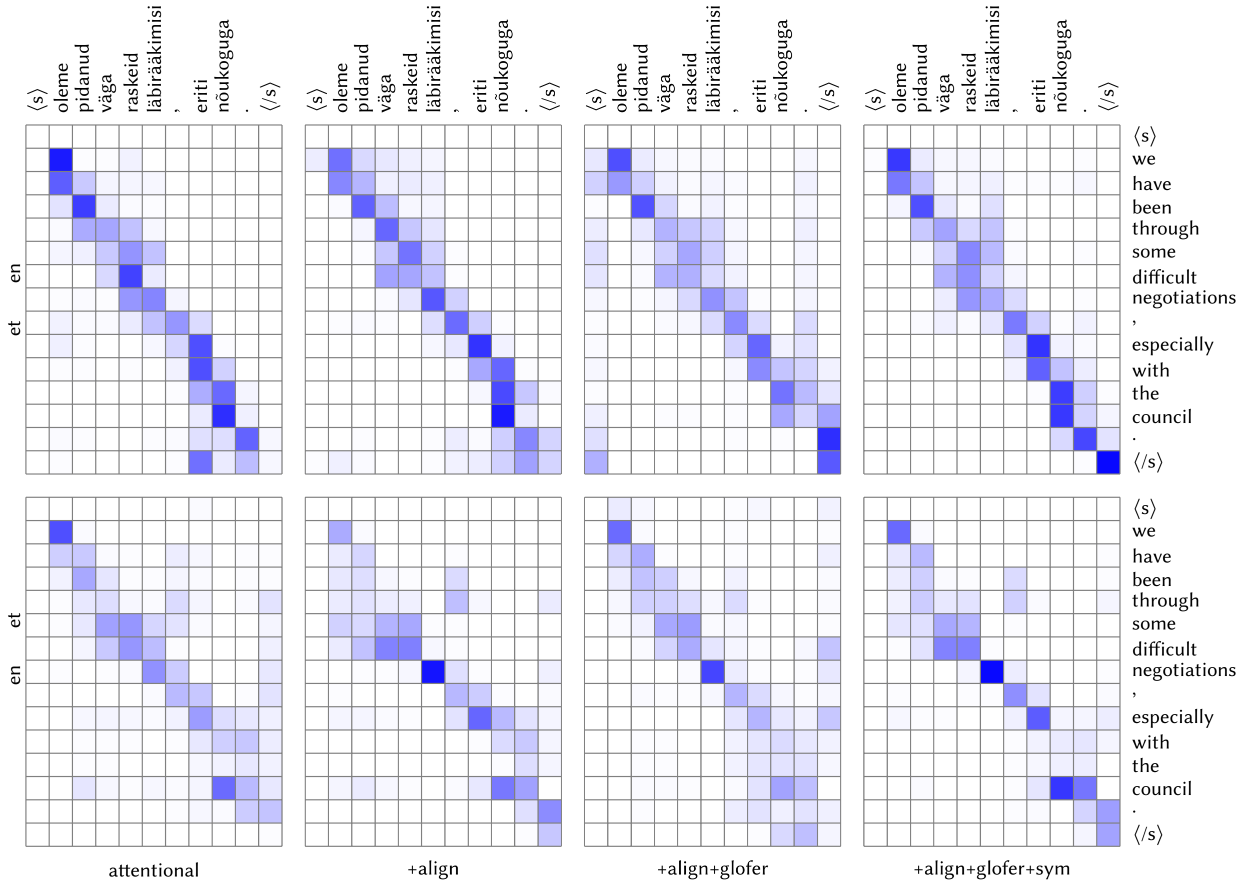

Figure 4: Example development sentence, showing the inferred attention matrix for various models for Et ↔ En. Rows correspond to the translation direction and columns correspond to different models: attentional, with alignment features (+align), global fertility (+glofer), and symmetric joint training (+sym). Darker shades denote higher values and white denotes zero.

pre). In this case, it is refining an already excellent model from which reliable global fertility estimates can be obtained. This finding is consistent with the other languages, see Figure 3 which shows typical learning curves of different variants of the attentional model. Note that when global fertility is added to the vanilla attentional model with alignment features, it significantly slows down training as it limits exploration in early training iterations, however it does bring a sizeable win when used to fine-tune a pre-trained model. Finally, the bilingual symmetry also helps to reduce the perplexity scores when used with the alignment features, however, does not combine well with global fertility (+align+sym+glofer-pre). This is perhaps an unsurprising result as both methods impose a often-times similar regularising effect over the attention matrix.

Figure 4 illustrates the different attention matrices inferred by the various model variants. Note the difference between the base attentional model and its variant with alignment features ('+align'), where more weight is assigned to diagonal and 1-to-many alignments. Global fertility pushes more attention to the sentinel symbols ⟨s⟩ and ⟨/s⟩. Determiners and prepositions in English show much lower fertility than nouns, while Estonian nouns have even higher fertility. This accords with Estonian morphology, wherein nouns are inflected with rich case marking, e.g., *nõukoguga* has the cogitative *-ga* suffix, meaning 'with', and thus translates as several English words (*with the council*). The right-most column corresponds to joint symmetric training, with many more confident attention values especially for consistent 1-to-many alignments (*difficult* in English and *raskeid* in Estonian, an adjective in partitive case meaning *some difficult*).

Lang. Pair	Zh-En	Ru-En	Et-En	Ro-En
Enc-Dec	5.35	61.9	18.2	10.3
Attentional	4.77	41.7	12.8	6.62
Our Work	**4.31**	**39.9**	**11.8**	**5.89**
Lang. Pair	En-Zh	En-Ru	En-Et	En-Ro
Enc-Dec	8.60	67.3	31.4	11.5
Attentional	7.49	43.0	19.4	7.30
Our Work	**6.24**	**40.6**	**17.0**	**6.35**

Table 3: Perplexity on the test sets for the two translation directions. Our work includes: bidirectional LSTM attentional model combined with positional bias, Markov, local fertility, and global fertility (pre-trained setting).

4.2 Experimental Results

The perplexity results of the neural models for the two translation directions across the four language pairs are presented in Table 3. In all cases, our work achieves lower perplexities compared to the vanilla attentional model and the encoder-decoder architecture, owing to the linguistic constraints. We also obtained similar patterns of improvements when decoding, using a greedy decoding strategy, as shown in Table 4. The exception was for en→ru, where the addition of the global fertility (in addition to the other aligment features) was detrimental, resulting in a decrease in BLEU score (5.94→5.26). This may be due to highly noisy nature of the web text corpus of Russian-English language pair, compared to the much cleaner sources for the other language pairs.

Greedy decoding does not appear to be competitive for neural models trained on small parallel corpora, not reaching the level of a phrase-based baseline (see Table 5). Despite this, however, these models still provide substantial gains when used for reranking (as shown in Table 5) for translating into English from the other four languages. We compare reranking settings using the log probabilities produced by our model as additional features[10] vs. using log probabilities from the vanilla attentional model and the encoder-decoder. The re-rankers based on our model are significantly better than the rest for Chinese and Estonian, and on par with the other for Russian and Romanian→English. In all cases our model has performance at least 1 BLEU point better than the baseline phrase-based system. It is worth noting that for Chinese-English, our re-ranker leads to a substantial increase of almost 3 BLEU points.

Lang. Pair	Zh-En	Ru-En	Et-En	Ro-En
Enc-Dec	17.4	3.63	12.5	21.2
Attentional	29.9	8.11	19.7	33.0
Our Work	**31.56**♠	**9.14**♠	**20.44**♠	**34.16**♠
Lang. Pair	En-Zh	En-Ru	En-Et	En-Ro
Enc-Dec	14.6	2.08	7.97	16.6
Attentional	20.9	5.26	12.5	28.1
Our Work	**23.45**♠	5.26	**13.40**♠	**30.07**♠

Table 4: BLEU scores on the test sets for the two translation directions, using greedy decoding. **bold:** Best performance, ♠: Statistically significantly better than Attentional.

Lang. Pair	Zh-En	Ru-En	Et-En	Ro-En
Phrase-based	40.63	18.70	31.99	45.21
Enc-Dec	40.41	18.83	32.20	45.36
Attentional	41.16	**19.79**	32.78	46.83
Our Work	**43.50**♠	19.73	**33.26**♠	**46.88**

Table 5: BLEU scores on the test sets for re-ranking. **bold:** Best performance, ♠: Statistically significantly better than Attentional.

5 Related Work

Kalchbrenner and Blunsom (2013) were the first to propose a full neural model of translation, using a convolutional network as the source encoder, followed by an RNN decoder to generate the target translation. This was extended in Sutskever et al. (2014), who replaced the source encoder with an RNN using a Long Short-Term Memory (LSTM) and leveraged the last hidden RNN states as source context for generating the output. Inspired by this, Bahdanau et al. (2015) introduced the notion of "attention" to the model, whereby the source context can dynamically change during the decoding process to attend to the most relevant parts of the source sentence. Further, Luong et al. (2015) refined the attention mechanism to be more local, by constraining attention to a text span, whose words' representations are averaged.

Similar in spirit to our work, recent research has proposed different ways of leveraging the attention history to incorporate alignment structural biases. (Luong et al., 2015) made use of the attention vector of the previous position when generating the attention vector for the next position. Feng et al. (2016)

[10] We include two features: the normalised log-probability of the translation, evaluated in both translation directions.

added another recurrent structure for the attention mechanism to enhance its memorization capabilities and capture long-range dependencies between the attention vectors. Tu et al. (2016) proposed a coverage vector to keep track of the attention history, hence refining future attentions. Finally, Cheng et al. (2015) proposed a similar agreement-based joint training for bidirectional attention-based neural machine translation, and showed significant improvements in BLEU for the large data French↔English translation.

6 Conclusion

We have shown that the attentional model of translation does not capture many well known properties of traditional word-based translation models, and proposed several ways of imposing these as structural biases on the model. We show improvements across several challenging language pairs in a low-resource setting, as well as in perplexity, translation and reranking evaluations. In future work we intend to investigate the model performance on larger-scale datasets, and incorporate further linguistic information, such as morphological representations.

Acknowledgements

The work reported here was started at JSALT 2015 in UW, Seattle, and was supported by JHU via grants from NSF (IIS), DARPA (LORELEI), Google, Microsoft, Amazon, Mitsubishi Electric, and MERL. Dr Cohn was supported by the ARC (Future Fellowship).

References

Alexandra Antonova and Alexey Misyurev. 2011. Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144.

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

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

Yong Cheng, Shiqi Shen, Zhongjun Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Agreement-based joint training for bidirectional attention-based neural machine translation. In *arXiv: 1512.04650 [cs.CL]*.

K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. 2014. On the properties of neural machine translation. In *arXiv:1409.1259 [cs.CL]*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

S. Feng, S. Liu, M. Li, and M. Zhou. 2016. Implicit Distortion and Fertility Models for Attention-based Encoder-Decoder NMT Model. *ArXiv e-prints*, January.

Kuzman Ganchev, João V. Graça, and Ben Taskar. 2008. Better alignments = better translations? In *Proceedings of ACL-08: HLT*, pages 986–993, Columbus, Ohio, June. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, October.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *IWSLT*, pages 68–75.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL Interactive Poster and Demonstration Sessions*, pages 177–180.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86. AAMT.

Tomer Levinboim, Ashish Vaswani, and David Chiang. 2015. Model invertibility regularization: Sequence alignment with or without parallel data. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 609–618, Denver, CO.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 104–111, New York, NY.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, Kyoto, Japan, October.

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

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems (NIPS)*, pages 3104–3112, Montréal.

Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li. 2016. Modeling Coverage for Neural Machine Translation. *ArXiv e-prints*, January.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 836–841.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2048–2057.