Continuous Space Reordering Models for Phrase-based MT

Nadir Durrani *Fahim Dalvi*

Qatar Computing Research Institute – HBKU

{ndurrani,faimaduddin}@qf.org.qa

Abstract

Bilingual sequence models improve phrase-based translation and reordering by overcoming phrasal independence assumption and handling long range reordering. However, due to data sparsity, these models often fall back to very small context sizes. This problem has been previously addressed by learning sequences over generalized representations such as POS tags or word clusters. In this paper, we explore an alternative based on neural network models. More concretely we train neuralized versions of lexicalized reordering [1] and the operation sequence models [2] using feed-forward neural network. Our results show improvements of up to 0.6 and 0.5 BLEU points on top of the baseline German→English and English→German systems. We also observed improvements compared to the systems that used POS tags and word clusters to train these models. Because we modify the bilingual corpus to integrate reordering operations, this allows us to also train a *sequence-to-sequence* neural MT model having explicit reordering triggers. Our motivation was to directly enable reordering information in the encoder-decoder framework, which otherwise relies solely on the `attention model` to handle long range reordering. We tried both coarser and fine-grained reordering operations. However, these experiments did not yield any improvements over the baseline Neural MT systems.

1. Introduction

Source-target bilingual sequence models have been used successfully as feature in phrase-based SMT [3, 2]. They are based on minimal translation units, and overcome independence assumption by handling non-local dependencies across phrasal boundaries, thus providing better translation and reordering mechanism. Such models however suffer from data sparsity and fall back to very small context sizes during test time. This shortcoming is addressed by learning factored models [4, 5, 6], learned over POS and morphological tags or using word classes [7, 8, 9].[1]

An alternative way to address data sparsity and learn better generalizations is to use continuous representations. Neural networks (NN) have shown success in Statistical Machine translation with n-best re-ranking [12, 13] or directly as a feature [14, 15] used during decoding. More recently, attention-based encoder-decoder Recurrant Neural Network (RNN) model [16], which trains a single large neural network, has emerged as the new state-of-the-art in MT.

In this work, we neuralize two commonly used reordering models namely lexicalized reordering [1] and the operation sequence model (OSM) [2] and integrate them as feature in phrase-based MT. We convert word-aligned bi-text into a sequence of operations through a deterministic algorithm (See Algorithm 1 in [17]), the resulting vocabulary and number of model parameters can become very large. A model trained on such representation may suffer from data sparsity. To overcome this, we separate the streams of source and target sequences and concatenate them to simulate the jointness. A feed-forward neural network is then trained on such concatenated n-gram sequences.

The OSM model exhibit very rich reordering operations varying from `Insert GAP` to `JUMP Forward` and `JUMP Backward` to multiple open gaps which may be hierarchically created. In an alternative method, we replace complicated reordering operations with `Monotonic`, `Swap and Discontinuous` operations, and train a neural model with coarser tags. This model is similar to the lexicalized reordering model, however much richer as it is conditioned on longer source-target contextual history and also previous reordering decisions.

We experimented with German-to-English and English-to-German language pairs. German is syntactically divergent from English and also exhibit very rich morphology, thus prone to data sparsity. These are the two problems we are addressing in this work. Our results show improvements of up to +0.6 and +0.5 BLEU points in German-to-English and English-to-German baselines respectively. We also demonstrated that neuralized OSM model performed better than the ones trained on POS tags and word-clusters. The neuralized OSM model outperformed the simpler lexicalized variant, although only slightly.

While training the Neural OSM (and Neural lexicalized reordering model) we embed reordering information in form of operations in the training corpus. This also allows us to train `sequence-to-sequence` neural MT system, where the target side is conditioned on both lexical and reordering states. [18] recently showed that integrating structural bias such as `Position bias`, i.e. relative positions of a source and corresponding target word, improves the attention mechanism. We tried to replicate this effect by i)

[1] as obtained during GIZA training [10] or using brown clusters [11]

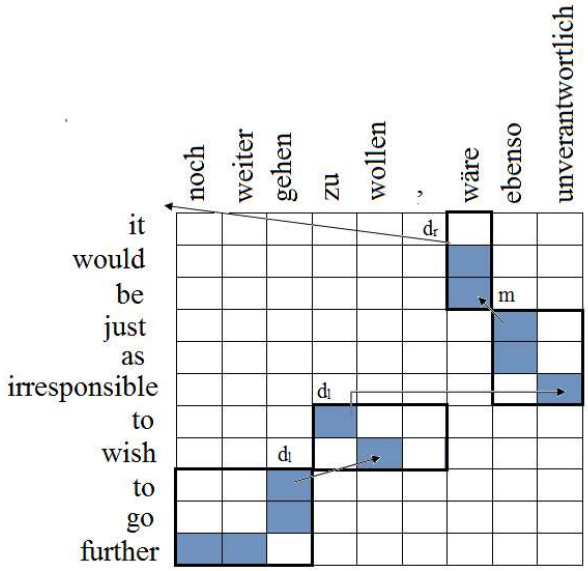

Figure 1: Lexical Reordering Models [20]: m = monotonic, s = swap, d = discontinuity (l: left, r: right)

linearizing the source to be in the same order as the target using word-alignments and ii) incorporating reordering states. Our motivation was that such reordering triggers will aid the attention model to better handle reordering. However, our results did not yield any improvements.

The remainder of this paper has been organized as follows: Section 2 describes the operation sequence and the lexicalized reordering model. We then present the neuralized versions of these models. Sections 3 and 4 describe our experimental setup and discusses the results. Section 5 gives account on related work and Section 6 concludes the paper.

2. Reordering Models

In this section we briefly revisit the two commonly used reordering models in the phrase-based Moses [19] namely the lexicalized reordering model and the operation sequence model. We then describe our neural versions of these models.

2.1. Lexicalized Reordering Model

The lexicalized reordering model originally proposed by [1] is the defacto reordering model used in phrase-based SMT (PBSMT). The idea is to learn orientation of a phrase w.r.t to previous phrase (or the last word of the previous phrase). An orientation could be one of the three reordering operations namely **M**`onotonic`, **S**`wap`, **D**`iscontinuous`. If the source phrases $F_{a(j-1)}$ and $F_{a(j)}$[2] are adjacent and in the same order as the target phrases E_{j-1} and E_j, the orientation is `Monotonic`. If they are in the opposite order of E_{j-1} and E_j, then the orientation is `Swap`, otherwise it is `Discontinuous`. See Figure 1 for illustration. For each phrase, we compute its probability of being reordered with the orientations $o = M, S, D$ as below:

$$p_r(o|F_{a(j)}, E_j) = \frac{count(o, F_{a(j)}, E_j)}{count(F_{a(j)}, E_j})$$

Improved versions [20, 21] have been subsequently integrated into Moses toolkit. The former computes orientation only based on the last word of the previous phrase, rather than the entire phrase and the latter, hierarchically combines all previous phrases to compute the probability. In our work, we will compute orientation based on previous source word, but condition on n previous source-target units. This is because our model is based on minimal translation units [3] and does not contain phrasal boundaries.

2.2. Operation Sequence Model

The operation sequence model (OSM) converts aligned bilingual corpus into a sequence of operations using a deterministic algorithm. An operation is either joint source-target lexical generation, or a reordering operation such as `Insert Gap` or `Jump Forward` or `Backward` to a specific open gap. A Markov model is estimated from the resulting operation corpus. More formally a bilingual sentence pair (T, S) and its word-alignment A is transformed deterministically to a heterogeneous sequence of translation and reordering operations $(o_1, o_2, \ldots, o_J)$. A 5-gram model is then learned over these sequences:

$$P_{osm}(T, S) \approx \prod_{j=1}^{J} P(o_j|o_{j-n+1}...o_{j-1})$$

The operation sequence for the example shown in Figure 1 according to the algorithm described in the original paper is given below:

Generate Target Only (it) – Insert Gap – Generate (wäre, would be) – Generate (ebenso, just as) – Generate (unverantwortlich, irresponsible) – Jump Back (1) – Insert Gap – Generate (zu, to) – Generate (wollen, wish) – Generate Source Only (,) – Jump Back (1) – Insert Gap – Generate (gehen, to go) – Jump Back (1) – Generate (noch weiter, further)

The OSM is trained on minimal translation units (MTUs) and does not adhere to phrasal boundaries. Access to joint source target information enables it to better handle long distance dependencies. The jumps and gap operations allow OSM to learn more complex reordering patterns. However, due to data sparsity it is impossible to observe all possible reordering patterns during the training. The model therefore falls back to very small context sizes. Earlier work has addressed this problem by estimating estimating the bilingual sequence models on POS tags or word clusters [22, 5, 6].

[2]The mapping function $a(j)$ aligns the target phrase E_j to the source phrase F_i, where $F_i = F_{a(j)}$.

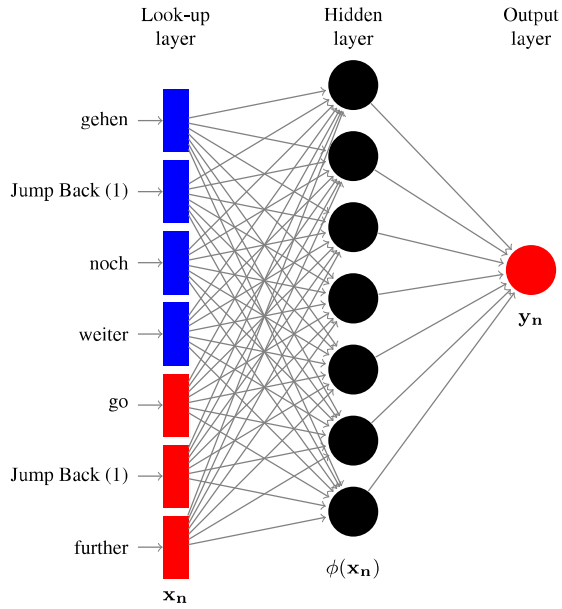

Figure 2: Neural OSM model where we use 3-gram target words and a source context window of size 4. For illustration, the output y_n is shown as a single categorical variable (scalar) as opposed to the traditional one-hot vector representation.

2.3. Neural Reordering Models

In this paper we take a different approach to address the problem of data sparsity by training the model using a feed forward neural network. Below we present the proposed neural versions of the OSM and lexicalized reordering models.

2.3.1. Neural Operation Sequence Model

A straight forward way is to build a neural language model using the generated sequences of operations. However, because of the joint nature of the model, the vocab size becomes quadratic ($M\times$N) causing severe data sparsity. A way to alleviate this problem is to separate out source and target streams and concatenate them to form history. See Table 1 for mapping operations into separate streams of source and target operations. Here are the considerations that we made: i) When a source or target word is unaligned (`Generate Source Only (Y)` or `Generate Target Only (X)` operations), we don't append anything on the other side, ii) Whenever there is a reordering operation (`Insert Gap/Jump Forward/Jump Back (N)`) we append it on both sides, iii) We replace source words on both sides for the `Generate Self` operation, iv) Multi-word source and target cepts are collapsed together even if they appear in a different order in the original sequence, v) Note that source-side is now reordered to be order of target just as in the original model. We generate separate streams of source and target operation and then concatenate them to train the neural model. Let $s_o = s_{o_1}, s_{o_2}...s_{o_n}$ and $t_o = t_{o_1}, t_{o_2}...t_{o_m}$ be streams of source and target operations, the model is defined as:

$$P(T,S) \approx \prod_{j=1}^{J} P(t_{o_j}|t_{o_{j-n+1}}...t_{o_{j-1}}, s_{o_j}...s_{o_{j-m+1}})$$

where m and n are the source and target word histories which we concatenate to form input to the neural network. As exemplified in Figure 2, this is essentially an $(m+n)$-gram neural network LM (NNLM) originally proposed by [23]. Each input word i.e. source or target vocabulary word or a reordering operation in the context is represented by a D dimensional vector in the shared look-up layer $L \in \mathbb{R}^{|V_i|\times D}$ where V_i is the input vocabulary.[3] The look-up layer then creates a context vector $\mathbf{x_n}$ representing the context words of the $(m+n)$-gram sequence by concatenating their respective vectors in L. The concatenated vector is then passed through non-linear hidden layers to learn a high-level representation, which is in turn fed to the output layer. The output layer has a `softmax` activation over the output vocabulary V_o of target words. Formally, the probability of getting k-th word in the output given the context $\mathbf{x_n}$ can be written as:

$$P(y_n = k|\mathbf{x_n}, \theta) = \frac{exp\ (\mathbf{w}_k^T \phi(\mathbf{x_n}))}{\sum_{m=1}^{|V_o|} exp\ (\mathbf{w}_m^T \phi(\mathbf{x_n}))} \quad (1)$$

where $\phi(\mathbf{x_n})$ defines the transformations of $\mathbf{x_n}$ through the hidden layers, and $\mathbf{w}_k$ are the weights from the last hidden layer to the output layer. For notational simplicity, henceforth we will use $(\mathbf{x_n}, y_n)$ to represent a training sequence. By setting m and n to be sufficiently large, neural OSM can capture long-range cross-lingual dependencies between words, while still overcoming the data sparseness issue by virtue of its distributed representations (i.e., word vectors).

2.3.2. Neural Lexicalized Reordering Model

We train the neural lexicalized reordering model in the same manner as that of the Neural OSM model. Traditional lexicalized reordering models use `Monotonic, Swap and Discontinuous`. We retained the `Swap (SW)` operation and divided the `Discontinuous (D)` category into `Forward Discontinuity (FD)` and `Backward Discontinuity (BD)` following [24]. We also removed the `Monotonic` orientation from the generation as it is obvious that words flow monotonically when there is no reordering. This is also done similarly in the OSM generation. Again like the Neural OSM generation, the reordering tags are split across both the streams. See Table 1 for the sample generation (last 2 columns).

Note that this model is not exactly the neural version of the lexicalized reordering in which the task is just to predict orientation/reordering decision (`Monotonic, Swap, Discontinuous`) based on previous source-target word (or phrase). Here we are trying to score the entire sequence

[3] Note that L is a model parameter to be learned.

Operations	Source Stream	Target Stream	Source Stream	Target Stream
Generate Target Only (it)	it		it	
Insert Gap	Insert Gap	Insert Gap	Jump Fwd	FD
Generate (wäre, would be)	wäre	would be	wäre	would be
Generate (ebenso, just as)	ebenso	just as	ebenso	just as
Generate (unverantwortlich, irresponsible)	unverantwortlich	irresponsible	unverantwortlich	irresponsible
Jump Back (1)	Jump Back (1)	Jump Back (1)	BD	BD
Insert Gap	Insert Gap	Insert Gap		
Generate (zu, to)	zu	to	zu	to
Generate (wollen, wish)	wollen	wish	wollen	wish
Generate Source Only (,)	,		,	
Jump Back (1)	Jump Back (1)	Jump Back (1)	BD	BD
Insert Gap	Insert Gap	Insert Gap		
Generate (gehen, to go)	gehen	to go	gehen	to go
Jump Back (1)	Jump Back (1)	Jump Back (1)	BD	BD
Generate (noch weiter, further)	noch weiter	further	noch weiter	further

Table 1: Operation Sequences and corresponding streams for Neural OSM and Lexicalized RM training

which contains both lexical (word generation) and reordering choices. The task is to find most probable sequence of lexical and reordering decisions. The difference compared to the OSM is the granularity of the reordering tags. In this model, we just have one reordering decision per lexical generation. In the OSM model, the model can have very complex sequence of reordering operations in between adjacent lexical generations. A more accurate version of the neural lexicalized reordering is described in [25]. They cast it as a classification problem, and use a continuous space representation treating a phrase as a dense real-valued vector. But unlike traditional model, they condition reordering probabilities also on the words of previous phrase to capture longer dependencies. This is similar to our work, except that our context information can go even beyond previous phrases and previous reordering decisions are also part of the context.

2.3.3. Neural Lexical Sequence Model

In this variation, we simply remove the reordering operations from the sequences and train the neural model only on the lexical sequences. This allows us to study how much of the improvement is obtained due reordering triggers integrated within these lexical sequences versus addressing sparsity by learning generalized representations. However note that such a lexical sequence model can still be considered a reordering model because the source was pre-ordered (or linearized) based on target (See Table 1) and generated in the target order. This model is based on the tuple sequence model [26] and several neural variants of it are presented in [13]. Another variation is presented in [15], but rather than pre-ordering the source, they select m neighboring word on the left and right sides of the source word s_i that is aligned to the target word t_i being modeled.

2.3.4. Decoding

We integrate these models as a feature in phrase-based decoding. Word alignments for the current phrase along with the history of previously generated operations are used to generate a new sequence of (lexical and reordering) operations. This sequence is then scored to give probability of the hypothesized phrase.

3. Experiments

3.1. Training Data

We experimented with German↔English language pairs using the data made available for the International Workshop on Spoken Language Translation (IWSLT'14). The data contains roughly 5M bilingual sentence pairs. We used only TED corpus [27] plus a subset of 800K parallel sentences from the rest of the parallel data to train the neural models.[4] We concatenated dev- and test-2010 for tuning and used test2011-2013 for evaluation.

3.2. MT Settings

We trained a Moses phrase-based system [19] following the settings described in [28]: maximum sentence length of 80, Fast-align [29] for word-alignments, an interpolated Kneser-Ney smoothed 5-gram language model [30], lexicalized reordering [31] and a 5-gram OSM model [2]. We used k-best batch MIRA [32] for tuning.[5] We trained alternative baselines by adding OSM models trained on POS and word clus-

[4]Training models on the entire data required roughly 18 days of wall-clock time (18 hours/epoch on a Linux Ubuntu 12.04.5 LTS running on a 16 Core Intel Xeon E5-2650 2.00Ghz and 64Gb RAM) on our machines. We ran one baseline experiment with all the data and did not find it better than the system trained on randomly selected subset of the data. In the interest of time, we therefore reduced the NN training to a subset (1M).

[5]All systems were tuned twice.

German-English				
System	test11	test12	test13	Avg.
Baseline	35.0	30.3	27.1	30.8
OSM_{pos}	35.3	30.5	27.1	31.0
OSM_{mkcls}	35.1	30.1	26.8	30.7
OSM_{neural}	35.8	31.5	27.0	31.4
Lex.reo_{neural}	35.5	31.1	27.2	31.3
Lex_{neural}	35.3	30.8	26.9	31.0
English-German				
Baseline	25.7	21.7	23.4	23.6
OSM_{pos}	25.9	21.9	23.8	23.9
OSM_{mkcls}	25.8	21.8	23.4	23.7
OSM_{neural}	26.1	22.1	24.2	24.1
Lex.reo_{neural}	26.1	22.4	23.7	24.1
Lex_{neural}	26.0	22.2	23.7	24.0

Table 2: Comparing performance of Neural Reordering Models against N-gram-based Models. Quality measured in cased-bleu [34]

ters (50) obtained by running `mkcls` [6]. We used LoPar for German and MXPOST tagger for English POS tags. We trained 7-gram models to enable wider context than the regular word-based models.

3.3. NN Training

We trained our neural reordering models using NPLM[6] toolkit [14] with the following settings. We used a target context of 6 words (including reordering operations) and a corresponding source window of 7 words (also including reordering operations), forming a joint stream of 14-grams for training. We restricted source and target side vocabularies to 20K and 40K most frequent words. We used an input embedding layer of 150 and an output embedding layer of 750. Only one hidden layer is used with a Noise Contrastive Estimation[7] or NCE [33]. Training was done using mini-batch size of 1000 and using 100 noise samples. All models were trained for 25 epochs.

3.4. Results

Table 2 compares the results for our neural reordering models against baseline containing traditional reordering models. The baseline system is equipped with lexicalized and OSM model trained over word forms using count-based/n-gram-based models. We see that adding OSM models trained over generalized representation such as POS tags help slightly (+0.2 BLEU improvement in DE-EN and +0.3 in EN-DE). Using word clusters instead of POS tags did not help as much.

The next set of rows show results when using neuralized OSM and Lexicalized reordering models. The neural OSM model gave an improvement of +0.6 and +0.5 in DE-EN and EN-DE pairs. Neuralized lexical reordering performed almost as good as the neural OSM model suggesting that fine-grained reordering tags and hierarchical jumps add little value. The lexical sequence model without reordering tags (last row) performed lower (in the DE-EN pair) showing that there is some value in integrating reordering tags[8] during generation. In the EN-DE pair the difference is insignificant showing that much of the gains are coming from addressing lexical sparsity and not better reordering.

4. Neural Machine Translation

Neural Machine Translation [16, 35] is quickly becoming the predominant approach to machine translation. Rather than modeling different linguistic aspects (lexical generation, reordering, fertility etc.) as feature components and tuning them to optimize BLEU, NMT is trained in an end-to-end fashion. Given a bilingual sentence pair, we first generate a vector representation of the source sentence using `encoder` and then map this vector to target sentence using a `decoder`. The long distance source and target contextual dependencies are modeled using recurrent neural networks (RNN) with bilingual Long Short Term Memory (LSTM) [36]. The attention model [16] serves as an alignment model which enables the decoder to focus on different parts of the source as it generates the target sentence. Unlike phrase-based decoding, the reordering window is not limited to a frame of 6 words. This allows NMT to capture very long range reordering like syntax-based models [37].

In this work, we tried to explore whether explicitly integrating reordering triggers into the RNN-based `encoder` and `decoder`, improve the performance of the attentional model. We use the training data generated earlier (to train the neural OSM models – See Table 1), to train the `sequence-to-sequence` NMT model. This allows the `decoder` to condition on both lexical and reordering states when generating the new target word, which itself can be a word or a reordering operation. Our motivation was that such reordering triggers and pre-ordering of source[9] might help the attention mechanism with its task.

Note that the target sequence and alignments are both latent variables during decoding, we need to predict the pre-ordered (or reordering augmented sequence). To do this, we additionally train a source→pre-ordered (or reordering augmented) source sequence using another `sequence-to-sequence` model.

[6] http://nlg.isi.edu/software/nplm/

[7] Training neural language model with backpropagation could be prohibitively slow because for each training instance, the softmax layer requires a summation over the entire output vocabulary. One way to avoid this repetitive computation is to use a Noise Contrastive Estimation of the loss function.

[8] We also tried variations with reordering tags either on source or target side. The current variation with tags on both sides worked best.

[9] Remember that we linearize the source based on target using word-alignments

German-English				
System	test11	test12	test13	Avg.
Baseline	33.9	29.2	27.5	30.2
OSM	32.2	27.6	25.6	28.5
Lex.reo	29.2	24.8	22.8	25.6
Lex	30.8	26.6	23.9	27.1

Table 3: Training NMT systems with pre-ordered data, with lexical reo. operations, OSM operations

German-English				
System	test11	test12	test13	Avg.
OSM	45.7	42.0	36.6	41.4
Lex.reo	48.0	45.2	43.2	45.5
Lex	52.0	50.8	49.3	50.7

Table 4: Source to pre-ordered (or reordering augmented) system

4.1. System Settings

We trained a 2-layered LSTM encoder-decoder with attention. We used `seq2seq-attn` implementation [38] with the following settings: word vectors and LSTM states have 500 dimensions, SGD with initial learning rate of 1.0 and rate decay of 0.5, and dropout of 0.3. The MT systems are trained for 20 epochs, and the model with best dev loss is used for extracting features for the classifier.

4.2. Results

Table 3 shows the results from training NMT systems from pre-ordered data and using reordering augmented data. No gains were observed compared to the baseline system. In fact there was significant drop in all cases. One reason for this drop could be inaccuracy in predicting pre-ordered (reordering augmented) sequences. This can be seen in the BLEU scores shown in Table 4.[10] [39] also found pre-ordering the source-side in Neural MT deteriorated system performance in Japanese↔English and Chinese↔English pairs. They conjectured that pre-ordering introduces noise in terms of word-order hindering the learning process more difficult.

5. Related Work

A significant amount of research has been carried to alleviate data sparsity when translating into or from morphologically rich languages. [4] integrated different levels of linguistic information as factors into the phrase-based translation model. The idea of translating to stems and then inflecting the stems in a separate step has been studied by several researchers [40, 41]. POS tags are used in bilingual sequence models to enable wider context by [5, 22, 6]. Several researchers used word clusters in training data to obtain smoother distributions and better generalizations [8, 7, 9]. [42] used factors and parallel back-offs to address the issue of data sparsity. Continuous space models are used earlier for n-best re-ranking or directly as a feature in phrase-based MT [12, 13, 43, 44, 15]. [45] recently proposed an LSTM recurrent neural reordering model which directly models word pairs and their alignment. However, because SMT decoder requires fixed history, it is only possible to use the feature in the n-best re-ranking.

A whole new paradigm based on deep neural network evolved as a parallel framework for machine translation [16, 35]. The RNN-based sequence-to-sequence model learns generalized representations to overcome data sparsity problems and learn long distance dependencies successfully. This is further enhanced by using sub-word [46] or character-based models [47] to address the OOV-word problem. [18] has recently shown that integrating structural biases based on relative positions and fertilities improves the attention mechanism. [48] and [49] used side-constraints i.e. adding suffix tag at the end of the source sentence or prefix tag in the beginning of the target sentence to control the behavior of the decoder i.e. politeness in the case of former and domain in the latter. Our work is similar in a sense that we are trying to add reordering constraints, forcing the decoder to produce a specific reordering pattern. However, our method did not yield any improvements.

6. Conclusion

Traditional reordering models in phrase-based system suffer from data sparsity. In this paper, we presented neuralized versions of these reordering models (the OSM and Lexicalized reordering models) and used them as a feature in Phrase-based SMT. Our evaluation on German-English language pairs showed an improvement of up to 0.6 BLEU points. We also demonstrated gains compared to the previous solution where these models are trained on parts-of-speech tags and word clusters, to address data sparsity and for better generalization. The code will be pushed to Moses toolkit.[11] We also tried our pre-ordered and reordering augmented training data to train sequence-to-sequence neural MT models, with a motivation to explicitly add reordering triggers in the encoder representation and aid the attention mechanism. However, our modification to the natural source order and integration of reordering symbols in the training data, did not yield improvement.

7. Acknowledgements

We would like to thank the anonymous reviewers for their useful feedback.

[10] The BLEU scores are computed using pre-ordered (or reordering augmented) references generated using word-alignments of original source-target evaluation sets.

[11] http://www.statmt.org/moses/

8. References

[1] C. Tillman, "A unigram orientation model for statistical machine translation," in *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds., Boston, Massachusetts, USA, May 2004, pp. 101–104.

[2] N. Durrani, A. Fraser, H. Schmid, H. Hoang, and P. Koehn, "Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT?" in *Proceedings of ACL 2013*, Sofia, Bulgaria, August 2013, pp. 399–405.

[3] H. Zhang, K. Toutanova, C. Quirk, and J. Gao, "Beyond left-to-right: Multiple decomposition structures for smt," in *Proceedings of NAACL-HLT 2013*, Atlanta, Georgia, June 2013, pp. 12–21.

[4] P. Koehn and H. Hoang, "Factored Translation Models," in *Proceedings of the Joint EMNLP-CoNLL 2007*, Prague, Czech Republic, June 2007, pp. 868–876.

[5] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Proceedings of the WMT-11*, Edinburgh, Scotland, July 2011, pp. 198–206.

[6] N. Durrani, P. Koehn, H. Schmid, and A. Fraser, "Investigating the usefulness of generalized word representations in smt," in *Proceedings of COLING 2014*, Dublin, Ireland, August 2014, pp. 421–432.

[7] V. Chahuneau, E. Schlinger, N. A. Smith, and C. Dyer, "Translating into morphologically rich languages with synthetic phrases," in *Proceedings of the EMNLP 2013*, Seattle, USA, October 2013, pp. 1677–1687.

[8] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving Statistical Machine Translation with Word Class Models," in *Proceedings of the EMNLP 2013*, Seattle, USA, October 2013, pp. 1377–1381.

[9] A. Bisazza and C. Monz, "Class-based language modeling for translating into morphologically rich languages," in *Proceedings of COLING 2014*, Dublin, Ireland, August 2014, pp. 1918–1927.

[10] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Comput. Linguist.*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[11] P. F. Brown, P. V. deSouza, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, pp. 467–479, 1992.

[12] H. Schwenk, "Continuous space translation models for phrase-based statistical machine translation," in *Proceedings of COLING 2012*, Mumbai, India, Dec 2012.

[13] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous space translation models with neural networks," in *Proceedings of the NAACL-HLT 2012*, Montréal, Canada, June 2012, pp. 39–48.

[14] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proceedings of the EMNLP 2013*, Seattle, USA, October 2013, pp. 1387–1392.

[15] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the ACL 2014*, Baltimore, USA, June 2014, pp. 1370–1380.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015. [Online]. Available: http://arxiv.org/pdf/1409.0473v6.pdf

[17] N. Durrani, H. Schmid, and A. Fraser, "A Joint Sequence Translation Model with Integrated Reordering," in *Proceedings of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*, Portland, OR, USA, 2011.

[18] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari, "Incorporating structural alignment biases into an attentional neural translation model," in *Proceedings of the NAACL-HLT 2016*, San Diego, California, June 2016, pp. 876–885.

[19] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the ACL 2007*, Prague, Czech Republic, 2007.

[20] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'05)*, Pittsburgh, PA, USA, 2005.

[21] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of EMNLP*, Honolulu, Hawaii, Oct 2008, pp. 848–856.

[22] J. M. Crego and F. Yvon, "Improving reordering with linguistically informed bilingual n-grams," in *In Proceedings of Coling 2010*, Beijing, China, August 2010, pp. 197–205.

[23] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[24] M. Nagata, K. Saito, K. Yamamoto, and K. Ohashi, "A clustered global phrase reordering model for statistical machine translation," in *Proceedings of COLING 2006*, Sydney, Australia, July 2006, pp. 713–720.

[25] P. Li, Y. Liu, M. Sun, T. Izuha, and D. Zhang, "A neural reordering model for phrase-based translation," in *Proceedings of COLING 2014*, Dublin, Ireland, August 2014, pp. 1897–1907.

[26] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà, "N-gram-Based Machine Translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[27] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th IWSLT Evaluation Campaign," *Proceedings of the IWSLT, Lake Tahoe, US*, 2014.

[28] A. Birch, M. Huck, N. Durrani, N. Bogoychev, and P. Koehn, "Edinburgh SLT and MT system description for the IWSLT 2014 evaluation," in *Proceedings of the 11th International Workshop on Spoken Language Translation*, ser. IWSLT '14, Lake Tahoe, CA, USA, 2014.

[29] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proceedings of NAACL'13*, 2013.

[30] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proceedings of the Sixth WMT*, Edinburgh, Scotland, UK, July 2011, pp. 187–197.

[31] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *Proceedings of the tenth Machine Translation Summit*, Phuket, Thailand, 2005.

[32] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of NAACL 2012*, Montréal, Canada, 2012, pp. 427–436.

[33] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of AISTATS*, ser. JMLR W&CP, vol. 9, 2010, pp. 297–304.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of ACL 2002*, Philadelphia, PA, USA, 2002, pp. 311–318.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer, "Scalable inference and training of context-rich syntactic translation models," in *Proceedings of the ACL 2006*, Sydney, Australia, July 2006.

[38] Y. Kim, "Seq2seq-attn," https://github.com/harvardnlp/seq2seq-attn, 2016.

[39] J. Du and A. Way, "Pre-reordering for neural machine translation: Helpful or harmful?" *prague bulletin of mathematical linguistics*, vol. 108, pp. 171–182, 2017.

[40] K. Toutanova, H. Suzuki, and A. Ruopp, "Applying Morphology Generation Models to Machine Translation," in *Proceedings of ACL-08: HLT*, Columbus, Ohio, June 2008, pp. 514–522.

[41] A. Fraser, M. Weller, A. Cahill, and F. Cap, "Modeling Inflection and Word-Formation in SMT," in *Proceedings of EACL 2012*, Avignon, France, April 2012, pp. 664–674.

[42] Y. Feng, T. Cohn, and X. Du, "Factored markov translation with robust modeling," in *Proceedings of CoNNL 2014*, Ann Arbor, Michigan, June 2014, pp. 151–159.

[43] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of EMNLP 2013*, Seattle, USA, October 2013, pp. 1700–1709.

[44] J. Gao, X. He, W.-t. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," in *Proceedings of ACL 2014*, Baltimore, Maryland, June 2014, pp. 699–709.

[45] Y. Cui, S. Wang, and J. Li, "Lstm neural reordering feature for statistical machine translation," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 977–982. [Online]. Available: http://www.aclweb.org/anthology/N16-1112

[46] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *arXiv preprint arXiv:1508.07909*, 2015.

[47] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware Neural Language Models," *arXiv preprint arXiv:1508.06615*, 2015.

[48] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings NAACL-HLT 2016*, San Diego, California, June 2016, pp. 35–40.

[49] C. Kobus, J. M. Crego, and J. Senellart, "Domain control for neural machine translation," *CoRR*, vol. abs/1612.06140, 2016. [Online]. Available: http://arxiv.org/abs/1612.06140