A Neural Reordering Model for Phrase-based Translation

Peng Li[†] **Yang Liu**[†] **Maosong Sun**[†] **Tatsuya Izuha**[‡] **Dakun Zhang**[*]

[†]State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Sci. and Tech., Tsinghua University, Beijing, China
pengli09@gmail.com, {liuyang2011,sms}@tsinghua.edu.cn

[‡]Toshiba Corporation Corporate Research & Development Center
tatsuya.izuha@toshiba.co.jp

[*]Toshiba (China) R&D Center
zhangdakun@toshiba.com.cn

Abstract

While lexicalized reordering models have been widely used in phrase-based translation systems, they suffer from three drawbacks: context insensitivity, ambiguity, and sparsity. We propose a neural reordering model that conditions reordering probabilities on the words of both the current and previous phrase pairs. Including the words of previous phrase pairs significantly improves context sensitivity and reduces reordering ambiguity. To alleviate the data sparsity problem, we build one classifier for all phrase pairs, which are represented as continuous space vectors. Experiments on the NIST Chinese-English datasets show that our neural reordering model achieves significant improvements over state-of-the-art lexicalized reordering models.

1 Introduction

Reordering plays a crucial role in phrase-based translation (Koehn et al., 2003; Och and Ney, 2004). While local reordering can be directly memorized in phrases, modeling reordering at a phrase level still remains a major challenge: it can be cast as a travelling salesman problem and proves to be NP-complete (Knight, 1999; Zaslavskiy et al., 2009).

The past decade has witnessed the rapid development of phrase reordering models (e.g., (Och et al., 2004; Tillman, 2004; Zens et al., 2004; Xiong et al., 2006; Al-Onaizan and Papineni, 2006; Koehn et al., 2007; Galley and Manning, 2008; Feng et al., 2010; Green et al., 2010; Bisazza and Federico, 2012; Cherry, 2013), just to name a few). Among them, *lexicalized reordering models* (Tillman, 2004; Koehn et al., 2007; Galley and Manning, 2008) have been widely used in practical phrase-based systems. Unlike the distance-based reordering model (Koehn et al., 2003) that only penalizes phrase displacements in terms of the degree of nonmonotonicity, lexicalized reordering models introduce reordering probabilities conditioned on the words of each phrase pair. They often distinguish between three orientations with respect to the previous phrase pair: *monotone*, *swap*, and *discontinuous*. As lexicalized reordering models capture the phenomenon that some words are far more likely to be displaced than others, they outperform unlexicalized reordering models substantially.

Despite their apparent success in statistical machine translation, lexicalized reordering models suffer from the following three drawbacks:

1. *Context insensitivity*. Lexicalized reordering models determine the orientations only depending on the words of current phrase pairs. In fact, a phrase pair usually has different orientations in different contexts. It is important to include more contexts to improve the expressive power of reordering models.

2. *Ambiguity*. Short phrase pairs, which are observed in the training data more frequently, usually have multiple orientations. We observe that about 92.4% of one-word Chinese-English phrase pairs are ambiguous. This makes it hard to decide which orientation should be properly used in decoding.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1897–1907, Dublin, Ireland, August 23-29 2014.

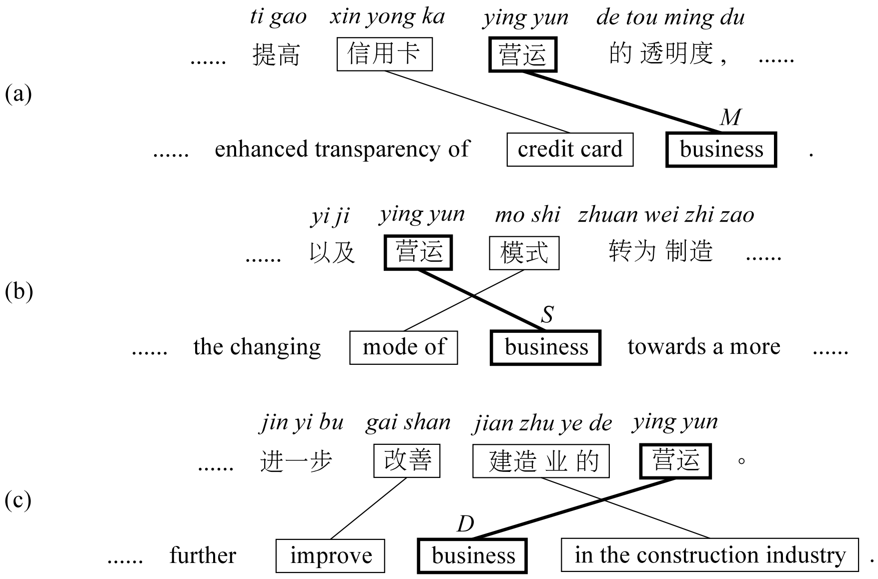
(a)
ti gao
xin yong ka
ying yun
de tou ming du
...... 提高 信用卡 营运 的 透明度， ......
M
...... enhanced transparency of credit card business .
(b)
yi ji
ying yun
mo shi
zhuan wei zhi zao
...... 以及 营运 模式 转为 制造 ......
S
...... the changing mode of business towards a more ......
(c)
jin yi bu
gai shan
jian zhu ye de
ying yun
...... 进一步 改善 建造 业 的 营运 。
D
...... further improve business in the construction industry .

Figure 1: Ambiguity in phrase reordering. The phrase pair ⟨“*yingyun*”, “business”⟩ is labeled with different orientations in different contexts: (a) *monotone*, (b) *swap*, and (c) *discontinuous*. Lexicalized reordering models use fixed probability distributions (e.g., 17.50% for M, 1.59% for S, and 80.92% for D) in decoding even though the surrounding contexts keep changing.

3. *Sparsity*. Lexicalized reordering models maintain a reordering probability distribution for each phrase pair. As most long phrase pairs that are capable of memorizing local word selection and reordering only occur once in the training data, maximum likelihood estimation can hardly train the models accurately.

In this work, we propose a neural reordering model for phrase-based translation. The contribution is twofold. Firstly, unlike conventional lexicalized reordering models, the neural reordering model conditions reordering probabilities on the words of both the current and previous phrase pairs. Including the words of previous phrase pairs significantly improves context sensitivity and reduces reordering ambiguity. Secondly, to alleviate the data sparsity problem, we build a neural classifier for all phrase pairs, which are represented as continuous space vectors. Experiments on the NIST Chinese-English datasets show that our neural reordering model achieves significant improvements over state-of-the-art lexicalized models.

2 Lexicalized Reordering Models

The lexicalized reordering models (Tillman, 2004; Koehn et al., 2007; Galley and Manning, 2008) have become the *de facto* standard in modern phrase-based systems. These models are called *lexicalized* because they condition reordering probabilities on the words of each phrase pair. Depending on the relationship between the current and previous phrase pairs, lexicalized reordering models often define *orientations* to classify different reordering patterns.

More formally, we use $\mathbf{f} = \{\tilde{f}_1, \dots, \tilde{f}_n\}$ to denote a sequence of source phrases, $\mathbf{e} = \{\tilde{e}_1, \dots, \tilde{e}_n\}$ to denote the phrase sequence on the target side, and $\mathbf{a} = \{a_1, \dots, a_n\}$ to denote the alignment between source and target phrases. A source phrase $\tilde{f}_{a_i}$ and a target phrase $\tilde{e}_i$ form a phrase pair. Lexicalized reordering models aim to estimate the conditional probability of a sequence of orientations $\mathbf{o} = \{o_1, \dots, o_n\}$:

$$P(\mathbf{o}|\mathbf{f}, \mathbf{e}, \mathbf{a}) = \prod_{i=1}^{n} P(o_i|\mathbf{f}, \tilde{e}_1, \dots, \tilde{e}_i, a_1, \dots, a_i) \tag{1}$$

where each o_i takes values over a set of predefined orientations. For simplicity, current lexicalized

model	source phrase length						
	1	2	3	4	5	6	7
$P(o_i\|\tilde{f}_{a_i}, \tilde{e}_i, a_{i-1}, a_i)$	92.74	54.01	24.09	14.40	10.78	8.47	6.95
$P(o_i\|\tilde{f}_{a_i}, \tilde{e}_i, \tilde{f}_{a_{i-1}}, \tilde{e}_{a_{i-1}}, a_{i-1}, a_i)$	21.72	5.22	2.63	1.48	0.98	0.67	0.54

Table 1: Percentages of phrase pairs that have multiple orientations. Including previous phrase pairs in modeling significantly reduces the reordering ambiguity for the M/S/D orientations. For example, while 92.74% of 1-word Chinese-English phrase pairs have multiple orientations observed in the training data, the ratio dramatically drops to 21.72% if the orientations are conditioned on both the current and previous phrase pairs.

reordering models use orientations conditioned only on a_{i-1} and a_i:

$$P(\mathbf{o}|\mathbf{f}, \mathbf{e}, \mathbf{a}) \approx \prod_{i=1}^{n} P(o_i|\tilde{f}_{a_i}, \tilde{e}_i, a_{i-1}, a_i) \tag{2}$$

The most widely used orientations are *monotone* (M), *swap* (S), and *discontinuous* (D): [1]

$$o_i = \begin{cases} M & \text{if } a_i - a_{i-1} = 1 \\ S & \text{if } a_i - a_{i-1} = -1 \\ D & \text{if } |a_i - a_{i-1}| \neq 1 \end{cases} \tag{3}$$

As lexicalized reordering models maintain a reordering probability distribution for each phrase pair, it is hard to accurately learn reordering probabilities for long phrase pairs that are usually observed only once in the training data. On the contrary, short phrase pairs that occur in the training data for many times tend to be ambiguous. For example, as shown in Figure 1, a Chinese-English phrase pair ⟨“*yingyun*”, “business”⟩ is observed to have different orientations in different contexts.

It is unreasonable to use fixed reordering probability distributions in decoding as the surrounding contexts keep changing. Previous study shows that considering more contexts into reordering modeling improves translation performance (Khalilov and Simaan, 2010). Therefore, we need a more powerful mechanism to include more contexts, resolve the reordering ambiguity, and reduce the data sparsity.

3 A Neural Reordering Model

3.1 The Model

Intuitively, conditioning reordering probabilities on the words of both the current and previous phrase pairs will significantly reduce both reordering ambiguity and context insensitivity. The new reordering model is given by

$$P(\mathbf{o}|\mathbf{f}, \mathbf{e}, \mathbf{a}) \approx \prod_{i=1}^{n} P(o_i|\tilde{f}_{a_i}, \tilde{e}_i, \tilde{f}_{a_{i-1}}, \tilde{e}_{i-1}, a_{i-1}, a_i) \tag{4}$$

where $\langle \tilde{f}_{a_{i-1}}, \tilde{e}_{i-1} \rangle$ is the previous phrase pair.

Including the previous phrase pairs improves the context sensitivity. For example, given a phrase pair ⟨“*yingyun*”, “business”⟩, its orientation is more likely to be *monotone* if it is preceded by a noun phrase pair such as ⟨“*xinyongka*”, “credit card”⟩. On the contrary, the probability of the *discontinuous* orientation is higher if the previous phrase pairs contain verbs such as ⟨“*gaishan*”, “improve”⟩. Therefore, the new model is capable of capturing the phenomenon that the orientation of a phrase pair depends on its surrounding contexts.

Another advantage of including previous phrase pairs is the reduction of reordering ambiguity. As shown in Table 1, 92.74% of 1-word Chinese-English phrase pairs have multiple orientations (i.e., M, S,

[1] There are many variants of lexicalized reordering models depending on the model type, orientation, directionality, language, and collapsing. See http://www.statmt.org/moses/?n=FactoredTraining.BuildReorderingModel for more details.

and D) observed in the training data. The ratio decreases with the increase of phrase length. In contrast, the new model is much less ambiguous (e.g., the ratio of ambiguous one-word phrase pairs dramatically drops to 21.72%) as it is conditioned on both the current and previous phrase pairs.

Unfortunately, including more contexts in modeling also increases the data sparsity. We observe that about 90% of reordering examples (i.e., the current and previous phrase pairs) are observed only once in the training data. As a result, it is more difficult to train lexicalized reordering models accurately using maximum likelihood estimation.

To alleviate the data sparsity problem, we use the following two strategies:

1. *Reordering as classification.* Instead of maintaining a reordering probability distribution for each phrase pair, we build a reordering classifier for all phrase pairs (Xiong et al., 2006; Li et al., 2013). This significantly reduces data sparsity by considering all occurrences of extracted phrase pairs as training examples. We find that $500,000$ reordering examples suffice to train a robust classifier (Section 4.5).

2. *Continuous space representation.* Instead of using a symbolic representation of phrases, we use a continuous space representation that treats a phrase as a dense real-valued vector (Socher et al., 2011b; Li et al., 2013). Consider two phrases “in London” and “in Centara Grand”. It is usually easy to predict the orientations of “in London” because it might be observed in the training data for many times. This is not the case for “in Centara Grand” as it might occur only once. However, if the two phrases happen to have very similar continuous space representations, “in Centara Grand” is likely to have a similar reordering probability distribution with “in London”.

To generate vector space representation for phrases, we follow Socher et al. (2011a) to use recursive autoencoders. Given two words w_1 and w_2, suppose their vector space representations are c_1 and c_2. The vector space representation p of the two-word phrase $\{w_1, w_2\}$ can be computed using a two-layer neural network:

$$p = g^{(1)}(W^{(1)}[c_1; c_2] + b^{(1)}) \tag{5}$$

where $[c_1; c_2] \in \mathbb{R}^{2n}$ is the concatenation of c_1 and c_2, $W^{(1)} \in \mathbb{R}^{n \times 2n}$ is a weight matrix, $b^{(1)}$ is a bias vector, and $g^{(1)}$ is an element-wise activation function.

In order to measure how well p represents c_1 and c_2, they can be reconstructed using another two-layer neural network:

$$[c'_1; c'_2] = g^{(2)}(W^{(2)}p + b^{(2)}) \tag{6}$$

where $c'_1 \in \mathbb{R}^n$ and $c'_2 \in \mathbb{R}^n$ are reconstructed vectors of c_1 and c_2, $W^{(2)} \in \mathbb{R}^{2n \times n}$ is a weight matrix, $b^{(2)} \in \mathbb{R}^n$ is a bias vector, and $g^{(2)}$ is an element-wise activation function. The reconstruction error can be measured by comparing c_1 and c_2 with c'_1 and c'_2. This process runs recursively in a bottom-up style to obtain the vector space representation of a multi-word phrase (Socher et al., 2011a). Socher et al. (2011a) find that minimizing the norms of hidden layers leads to the reduction of reconstruction error in an undesirable way. Therefore, we normalize p such that $||p||_2 = 1$.

Treating phrase reordering as a classification problem, we propose a neural reordering classifier that takes the current and previous phrase pairs as input. The neural network consists of four recursive autoencoders and a softmax layer. The input of the classifier are the previous phrase pair and the current phrase pair. Four recursive autoencoders are used to transform the four phrases (i.e., $\tilde{f}_{a_i}, \tilde{e}_i, \tilde{f}_{a_{i-1}}, \tilde{e}_{i-1}$) into vectors. Then, these vectors are fed to the softmax layer to predict reordering orientations. Note that the recursive autoencoders for the same language share with the same parameters. Our neural network is similar to that of Li et al. (2013). The major difference is that Li et al. (2013) need to compute vector space representation for variable-sized blocks ranging from words to sentences on the fly both in training and decoding. In contrast, we only need to compute vectors for phrases with up to 7 words in the training phase, which makes our approach simpler and more scalable to large data.

Formally, given the previous phrase pair $\langle \tilde{f}_{a_{i-1}}, \tilde{e}_{i-1} \rangle$, the current phrase pair $\langle \tilde{f}_i, \tilde{e}_i \rangle$ and the orientation o_i, the reordering probability is computed as

$$P(o_i | \tilde{f}_{a_i}, \tilde{e}_i, \tilde{f}_{a_{i-1}}, \tilde{e}_{i-1}, a_{i-1}, a_i) = g(W^o c(\tilde{f}_{a_i}, \tilde{e}_i, \tilde{f}_{a_{i-1}}, \tilde{e}_{i-1}) + b^o), \tag{7}$$

where W^o is a weight matrix, b^o is a bias vector, $c(\tilde{f}_{a_i}, \tilde{e}_i, \tilde{f}_{a_{i-1}}, \tilde{e}_{i-1})$ is the concatenation of the vectors of the four phrases. [2]

Following Och (2003), we use a linear model in our decoder with conventional features (e.g., translation probabilities and n-gram language model). The neural reordering model is incorporated into the discriminative framework as an additional feature.

3.2 Training

Training the neural reordering model involves minimizing the following two kinds of errors:

- *Reconstruction error*: It measures how well the computed vector space representations represent the input vectors. It is defined as the average reconstruction error of all the parent nodes in the trees formed during computing the vector space representation for all the phrases in the training data.

- *Classification error*: It measures how well the resulting classifier predicts the reordering orientations. It is defined as the average cross-entropy errors of all the training examples.

In our experiments, the objective function is a linear interpolation of the reconstruction error and the classification error.

Following Socher et al. (2011b), we use L-BFGS (Liu and Nocedal, 1989) to optimize the parameters. At the beginning of each iteration, a binary tree for each phrase is constructed using a greedy algorithm (Socher et al., 2011b). [3] With these trees fixed, the partial derivatives with respect to parameters are computed via the backpropagation through structures algorithm (Goller and Kuchler, 1996).

When optimizing the parameters of the softmax layer, the training procedure keeps the parameters of the recursive autoencoders and word embedding matrices fixed. The corresponding error function is the classification error as described above. We also use L-BFGS to optimize the parameters and the standard error backpropagation algorithm (Rumelhart et al., 1986) to compute the derivatives.

3.3 Decoding

As the vector space representation of a phrase is calculated based on all the words in the phrase, using the neural reordering model complicates the conditions for risk-free hypothesis recombination (Koehn et al., 2003). Therefore, many hypotheses are not likely to be recombined if the neural reordering model is directly integrated in decoding, making the decoder to only explore in a much smaller search space. [4] Therefore, we use Moses to generate search graphs and then use *hypergraph reranking* (Huang and Chiang, 2007; Huang, 2008) to find most probable derivations using the neural reordering model.

4 Experiments

4.1 Data Preparation

We evaluate our reordering model on Chinese-English translation. The training corpus consists of 1.23M sentence pairs with 32.1M Chinese words and 35.4M English words. A 4-gram language model was trained on the Xinhua portion of the English GIGAWORD corpus using KenLM (Heafield, 2011), which contains 398.6M words. We used the NIST 2006 MT Chinese-English dataset as the development set, and NIST 2002-2005, 2008 MT Chinese-English datasets as the test sets. Case-insensitive BLEU is used

[2] In practice, as suggested by Socher et al. (2011b), we feed the four average vectors of the vectors present in each recursive autoencoders to the softmax layer. Taking "resident population" as an example, there are three vectors in the binary tree used by the corresponding recursive autoencoder, denoted as $\hat{x}_1$, $\hat{x}_2$ and $\hat{x}_3$. The average vector is computed as $\bar{x} = \frac{1}{3}\sum_{i=1}^{3} \hat{x}_i$.

[3] As phrases in phrase-based translation are not necessarily syntactic constituents, we do not use parse trees in this work.

[4] Experimental results show that we can only achieve comparable performance with Moses by integrating neural reordering model directly in decoding.

Model	Orientation	MT06	MT02	MT03	MT04	MT05	MT08
distance	N/A	29.56	31.40	31.27	31.34	29.98	23.87
word	M/S/D	30.19	32.03	31.86	*32.09*	30.55	24.20
	left/right	30.17	31.98	31.52	31.98	30.19	24.30
phrase	M/S/D	30.24	32.35	31.85	32.00	*30.78*	24.33
	left/right	29.57	*32.64*	31.53	31.90	30.70	24.28
hierarchical	M/S/D	*30.46*	32.52	*31.89*	*32.09*	30.39	24.11
	left/right	30.03	32.13	31.59	31.91	30.21	*24.41*
neural	M/S/D	30.68	32.19	31.94	32.20	30.81	24.71
	left/right	**31.03****	**33.03****	**32.48****	**32.52****	**31.11***	**25.20****

Table 2: Comparison of distance-based, lexicalized, and neural reordering models in terms of case-insensitive BLEU-4 scores. "distance" denotes the distance-based reordering model (Koehn et al., 2003), "word" denotes the word-based lexicalized model (Tillman, 2004), "phrase" denotes the phrase-based lexicalized model (Koehn et al., 2007), "hierarchical" denotes the hierarchical phrase-based reordering model (Galley and Manning, 2008), and "neural" denotes our model. The "left" and "right" orientations only considers whether the current source phrase is on the left of the previous source phrase or not. We use "*" to highlight the result that is significantly better than the best baseline (highlighted in italic) at $p < 0.05$ level and "**" at $p < 0.01$ level. The neural model does not work well for the M/S/D orientations due to the non-separability problem (Section 4.3).

as the evaluation metric. As a trade-off between expressive power and computational cost, we set the dimension of the word embedding vectors to 25. [5] Both $g^{(1)}$ and $g^{(2)}$ are set to $\tanh(\cdot)$. The other hyperparameters are optimized via random search (Bergstra and Bengio, 2012).

4.2 Comparison of Distance-based, Lexicalized, and Neural Reordering Models

We compare three kinds of reordering models with increasing expressive power:

1. *distance-based model*: penalizing phrase displacements proportionally to the amount of nonmonotonicity (Koehn et al., 2003);

2. *lexicalized models*: conditioning the reordering probabilities on the current phrase pairs. The orientations can be determined with respect to words (Tillman, 2004), phrases (Koehn et al., 2007), or hierarchical phrases (Galley and Manning, 2008);

3. *neural model*: conditioning the reordering probabilities on both the current and previous phrase pairs.

For lexicalized and neural models, we further distinguish between two kinds of orientation sets: $\{$*monotone, swap, discontinuous*$\}$ and $\{$*left, right*$\}$. The *left/right* orientations only consider whether the current source phrase is on the left of the previous source phrase or not. Therefore, *swap* and *discontinuous-left* are merged into *left* while *monotone* and *discontinuous-right* into *right*.

All these reordering models are tested using Moses (Koehn et al., 2007), except that the neural model needs an additional hypergraph reranking procedure (Section 3.3). Implemented using Java, it takes the reranker 0.748 second to rerank a hypergraph on average.

Table 2 shows the case-insensitive BLEU-scores of distance-based, lexicalized, and neural reordering models on the NIST Chinese-English datasets. "distance" denotes the distance-based reordering model (Koehn et al., 2003), "word" denotes the word-based lexicalized model (Tillman, 2004), "phrase" denotes the phrase-based lexicalized model (Koehn et al., 2007), "hierarchical" denotes the hierarchical phrase-based reordering model (Galley and Manning, 2008), and "neural" denotes our model.

[5] We find that the dimensions of vectors do not have a significant impact on translation performance. For efficiency, we set the dimension to 25.

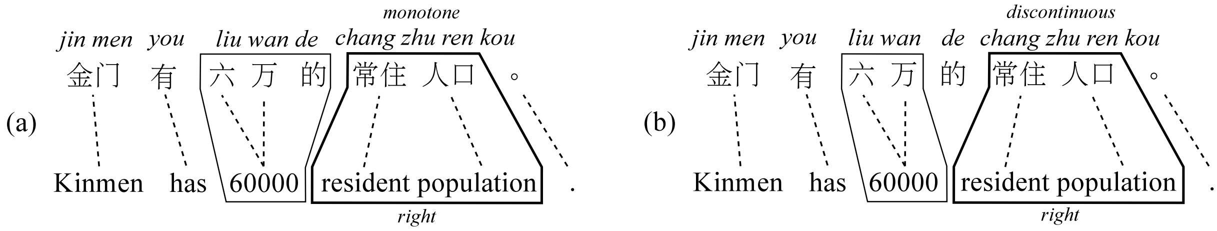
monotone
jin men you liu wan de chang zhu ren kou
金门 有 六 万 的 常住 人口 。
(a)
Kinmen has 60000 resident population .
right
discontinuous
jin men you liu wan de chang zhu ren kou
金门 有 六 万 的 常住 人口 。
(b)
Kinmen has 60000 resident population .
right

Figure 2: The non-separability problem for the neural reordering model. Given an aligned Chinese-English sentence pair, the unaligned Chinese word "*de*" makes a big difference in determining M/S/D orientations. In (a), "*de*" is included in the previous source phrase and thus the orientation is *monotone*. In (b), however, it is not included in the previous source phrase and the orientation is *discontinuous*. In our neural reordering model, "*liu wan de*" and "*liu wan*" have very similar vector space representations yet different orientations (i.e., M and D). In other words, training examples labeled with M, S, D are prone to be mixed with each other in the vector space. Therefore, it is difficult to find a hyperplane to separate M, S and D examples in the high-dimensional space.

We find that lexicalized reordering models obtain significant improvements over the distance-based model, which indicates that conditioning reordering probabilities on the words of the current phrase pairs does improve the expressive power. Our neural model using *left*/*right* orientations significantly outperforms all variants of lexicalized models. We use "*" to highlight the result that is significantly better than the best baseline (highlighted in italic) at $p < 0.05$ level and "**" at $p < 0.01$ level. This suggests that conditioning reordering probabilities on the words of current and previous phrase pairs is helpful for resolving reordering ambiguities and reducing context insensitivity.

4.3 The Non-Separability Problem

In Table 2, the neural model using the M/S/D orientations fails to outperform lexicalized models significantly. One possible reason is that the neural model suffers from the *non-separability problem* due to the M/S/D orientations.

As shown in Figure 2, given an aligned Chinese-English sentence pair, the unaligned Chinese function word "*de*" makes a big difference in determining M/S/D orientations. In (a), "*de*" is included in the previous source phrase and thus the orientation is *monotone*. In (b), however, "*de*" is not included in the previous source phrase and the orientation is *discontinuous*. In our neural reordering model, "*liu wan de*" and "*liu wan*" have very similar vector space representations yet different orientations (i.e., M and D). In other words, training examples labeled with M, S, D are prone to be mixed with each other in the vector space. Therefore, it is difficult to find a hyperplane to separate M, S and D examples in the high-dimensional space.

Fortunately, we find that using the *left*/*right* orientations can alleviate this problem. As the *left*/*right* orientations only consider whether the current source phrase is on the left of the previous source phrase or not, unaligned source words will not change orientations. For example, both Figure 2(a) and 2(b) are identified as the *right* orientation.

As a result, using *left*/*right* orientations in the neural reordering model not only has a higher classification accuracy (85%) over using the M/S/D orientations (69%), but also achieves higher BLEU scores on all NIST datasets systematically.

4.4 The Effect of Distortion Limit

Figure 3 shows the performance of the lexicalized model and our neural model with various distortion limits. The lexicalized model is the word-based model with M/S/D orientations. The neural model uses *left*/*right* orientations. The neural model consistently outperforms the lexicalized model, especially for large distortion limits. This finding suggests that the neural model is superior to lexicalized models in predicting long-distance reordering.

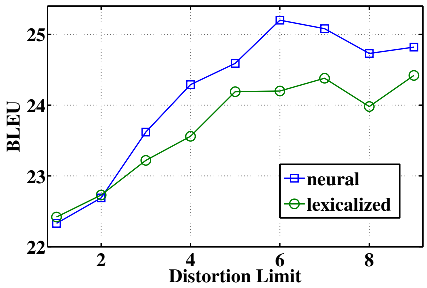

Figure 3: BLEU with various distortion limits.

# examples	Accuracy	BLEU
100,000	83.55	30.92
200,000	84.40	31.03
300,000	84.55	31.01
400,000	84.95	30.93
500,000	85.25	31.27
3,000,000	85.55	31.03

Table 3: Effect of training corpus size.

Vectors	MT06	MT02	MT03	MT04	MT05	MT08
ours	31.03	33.03	32.48	32.52	31.11	25.20
word2vec	30.44	32.28	32.00	32.07	30.24	24.54

Table 4: Comparison of neural reordering models trained based on word vectors produced by our model (ours) and word2vec (Mikolov et al., 2013).

4.5 The Effect of Training Corpus Size

Table 3 shows the classification accuracy and translation performance with various number of randomly sampled reordering examples for training the neural classifier. The classification accuracy and translation performance generally rise as the number of reordering example increases.[6] Surprisingly, both the classification accuracy and translation performance of using 500,000 reordering examples are close to using 3,000,000 reordering examples, suggesting that a relatively small amount of reordering examples are enough for training a robust classifier.

4.6 Learned Vector Space Representations

We randomly sampled 200,000 English phrases and found 999 clusters according to the vector space representations computed by recursive autoencoders using the k-means algorithm (MacQueen, 1967). The distance between two phrases is calculated by the Euclidean distance between their vector space representations.

Figure 4 shows 10 of the 999 clusters. An interesting finding is that phrase pairs that are close in the vector space share with similar reordering patterns rather than semantic similarity. For example, "by june 1" and "within the agencies" have similar distributions on the *left*/*right* orientations but are totally unrelated in terms of meaning. As a result, the vector representations of words trained using unlabeled data hardly helps in training the neural reordering model. Table 4 shows the results when we replace the word vectors of our model with those trained using word2vec (Mikolov et al., 2013). The recursive autoencoders and the classifier are retrained. The performance of the neural reordering model trained in this way drops significantly, which confirms our analysis.

5 Related Work

Reordering as classification is a common way to alleviate the data sparsity problem. Xiong et al. (2006) use a maximum entropy model to predict whether to merge two blocks in a straight or an inverted order in their ITG decoder. Nguyen et al. (2009) build a similar model for hierarchical phrase reordering models (Galley and Manning, 2008). Green et al. (2010) and Yahyaei and Monz (2010) predict finer-grained distance bins instead. Another direction is to learn sparse reordering features and create more flexible distributions (Cherry, 2013). Although these models are effective, feature engineering is a major challenge. In contrast, our neural reordering model is capable of learning features automatically.

[6] The reason why the BLEU scores oscillate slightly on the training set is that classification accuracy is not directly correlated with BLEU scores. Optimizing the neural reordering model directly with respect to BLEU score may further improve the performance. We leave this for future work.

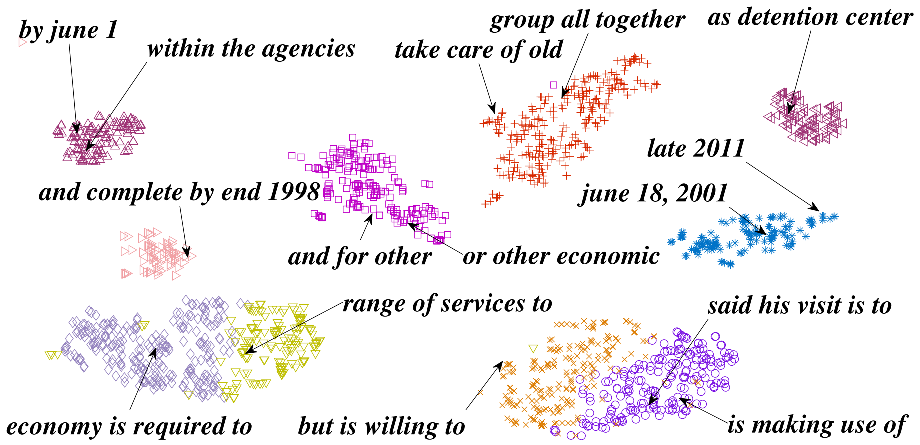

Figure 4: Phrase clusters as calculated by the Euclidean distance in the vector space. English phrases that have similar reordering probability distributions rather than similar semantic similarity fall into one cluster.

Along another line, n-gram-based models (Mariño et al., 2006; Durrani et al., 2011; Durrani et al., 2013) treat translation as Markov chains over minimal translation units (Mariño et al., 2006; Durrani et al., 2013) or operations (Durrani et al., 2011) directly. Although naturally leveraging both the source and target side contexts, these approaches still face the data sparsity problem.

Our work is closely related to Li et al. (2013). The major difference is that Li et al. (2013) need to compute vector space representation for variable-sized blocks ranging from words to sentences on the fly both in training and decoding. In contrast, we only need to compute vectors for phrases with up to 7 words in the training phase, which makes our approach simpler and more scalable to large data.

6 Conclusion

We have shown that surrounding context is effective for resolving reordering ambiguities in phrase-based models. As the data sparseness problem is the major challenge for using context in reordering models, we propose to use a single classifier based on recursive autoencoders to predict reordering orientations. Experimental results show that our neural reordering model outperforms the state-of-the-art lexicalized reordering models significantly and consistently across all the NIST datasets under various settings.

There are a few future directions we plan to explore. First, as the machine translation system and neural classifier are trained separately, the neural network training only has an indirect effect on translation quality. Jointly training the machine translation system and neural classifier is an interesting topic. Second, it is interesting to develop more efficient models to leverage larger contexts to resolve reordering ambiguities. Third, we plan to extend our work to other translation models such as syntax-based and n-gram based models (Mariño et al., 2006; Durrani et al., 2011; Durrani et al., 2013). Finally, as we cast phrase reordering as two-category classification problem (i.e, *left* vs. *right*), it is interesting to intersect structured SVM (Tsochantaridis et al., 2005) with neural networks to develop a large margin training algorithm for our neural reordering model.

Acknowledgements

This research is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (No. 61331013), the 863 Program (No. 2012AA011102), Toshiba Corporation Corporate Research & Development Center, and the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305.

Arianna Bisazza and Marcello Federico. 2012. Modified distortion matrices for phrase-based statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 478–487.

Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 22–31.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1045–1054.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can Markov models over minimal translation units help phrase-based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 399–405.

Yang Feng, Haitao Mi, Yang Liu, and Qun Liu. 2010. An efficient shift-reduce decoding algorithm for phrased-based machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 285–293.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of 1996 IEEE International Conference on Neural Networks (Volume:1)*, volume 1, pages 347–352.

Spence Green, Michel Galley, and Christopher D. Manning. 2010. Improved models of distortion cost for statistical machine translation. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 867–875.

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.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 586–594.

Maxim Khalilov and Khalil Simaan. 2010. Source reordering using maxent classifiers and supertags. In *Proceedings of The 14th Annual Conference of the European Association for Machine Translation*, pages 292–299.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Peng Li, Yang Liu, and Maosong Sun. 2013. Recursive autoencoders for ITG-based translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 567–577.

DongC. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528.

James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297.

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta R. Costa-jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Vinh Van Nguyen, Akira Shimazu, Minh Le Nguyen, and Thai Phuong Nguyen. 2009. Improving a lexicalized hierarchical reordering model using maximum entropy. In *Proceedings of The twelfth Machine Translation Summit (MT Summit XII)*.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004: Main Proceedings*, pages 161–168.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 318–362.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Andrew Y. Ng, and Christopher D. Manning. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of Advances in Neural Information Processing Systems 24*, pages 801–809.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011b. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004: Short Papers*, pages 101–104.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528.

Sirvan Yahyaei and Christof Monz. 2010. Dynamic distortion in a discriminative reordering model for statistical machine translation. In *Proceedings of the 7th International Workshop on Spoken Language Translation*, pages 353–360.

Mikhail Zaslavskiy, Marc Dymetman, and Nicola Cancedda. 2009. Phrase-based statistical machine translation as a traveling salesman problem. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 333–341.

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 205–211.