A Dependency-Based Neural Reordering Model for Statistical Machine Translation

Christian Hadiwinoto, Hwee Tou Ng
Department of Computer Science
National University of Singapore
{chrhad, nght}@comp.nus.edu.sg

Abstract

In machine translation (MT) that involves translating between two languages with significant differences in word order, determining the correct word order of translated words is a major challenge. The dependency parse tree of a source sentence can help to determine the correct word order of the translated words. In this paper, we present a novel reordering approach utilizing a neural network and dependency-based embeddings to predict whether the translations of two source words linked by a dependency relation should remain in the same order or should be swapped in the translated sentence. Experiments on Chinese-to-English translation show that our approach yields a statistically significant improvement of 0.57 BLEU point on benchmark NIST test sets, compared to our prior state-of-the-art statistical MT system that uses sparse dependency-based reordering features.

Introduction

In a machine translation (MT) system, determining the correct word order of translated words is crucial as word order reflects meaning. As different languages have different ordering of words, reordering of words is required to produce the correct translation output. Reordering in MT remains a major challenge for language pairs with a significant word order difference. Phrase-based MT systems (Koehn, Och, and Marcu 2003), which achieve state-of-the-art performance, generally adopt a reordering model based on the span of a phrase and the span of its adjacent phrase (Tillmann 2004; Koehn et al. 2005; Galley and Manning 2008).

Incorporating the dependency parse tree of an input (source) sentence is beneficial for reordering, as the dependency tree captures the relationships between words in a sentence, through the dependency relation label between two words. The dependency parse tree of a source sentence can be utilized in reordering integrated within phrase-based statistical MT (SMT), by defining dependency-based features in the SMT log-linear model (Chang et al. 2009; Hadiwinoto, Liu, and Ng 2016).

Recently, neural networks have been applied to natural language processing (NLP) to minimize feature engineering and to utilize continuous word representation. It has found application in MT reordering, applied in the reranking of translation output candidates (Li et al. 2014; Cui, Wang, and Li 2016) and in the pre-ordering approach (reordering the source sentence before translation) (de Gispert, Iglesias, and Byrne 2015; Miceli-Barone and Attardi 2015). Nevertheless, reordering integrated with translation has not benefited from the neural approach.

In this paper, we propose applying a neural network (NN) model in reordering integrated within translation. We apply our neural classifier on two words linked by a dependency relation link, either a head-child or sibling link, in order to predict if two words need to be swapped in the translation. The prediction is used to guide the decoding process of state-of-the-art phrase-based SMT.

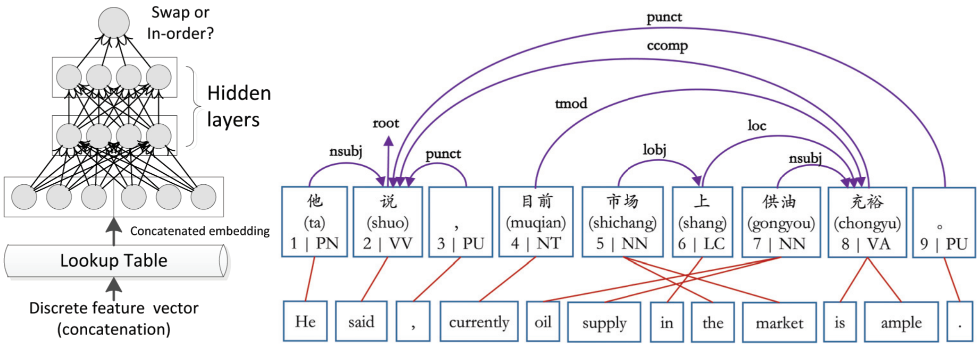
A Neural Classifier for Dependency-Based Reordering

We propose two neural classifiers, one to predict the correct order of the translated target words of two source words with a head-child relation, and the other for two source words with a sibling relation. Each binary classifier takes a set of features related to the two source words as its input and predicts if the translated words should be swapped (positive) or remain in order (negative).

Input Representation

The head-child classifier predicts the order of the translated words of a source word x_c and its head word x_h (where x_g is the head word of x_h) using the following input features:

- The head word x_h, its part-of-speech (POS) tag $T(x_h)$, and the dependency label $L(x_h)$ linking x_h to x_g
- The child word x_c, its POS tag $T(x_c)$, and the dependency label $L(x_c)$ linking x_c to x_h
- The signed distance $d(x_h, x_c)$ between the head and the child in the original source sentence, with the following possible values:
  - -2 if x_c is on the left of x_h and there is at least one other child between them
  - -1 if x_c is on the left of x_h and there is no other child between them
  - $+1$ if x_c is on the right of x_h and there is no other child between them



(a) (b)

Pair	Head			Child, if left			Child, if right			Dist.	Punct.	Label
	x_h	$T(x_h)$	$L(x_h)$	x_c	$T(x_c)$	$L(x_c)$	x_c	$T(x_c)$	$L(x_c)$	$d(x_h, x_c)$	$\omega(x_h, x_c)$	
(2, 1)	shuo	VV	root	ta	PN	nsubj	NULL	NULL	NULL	-1	0	0
(2, 8)	shuo	VV	root	NULL	NULL	NULL	chongyu	VA	ccomp	$+2$	1	0
(6, 5)	shang	LC	loc	shichang	NN	lobj	NULL	NULL	NULL	-1	0	1

(c)

Pair	Left child				Right child				Head		Punct.	Label
	x_l	$T(x_l)$	$L(x_l)$	$d(x_h, x_l)$	x_r	$T(x_r)$	$L(x_r)$	$d(x_h, x_r)$	x_h	$T(x_h)$	$\omega(x_l, x_r)$	
(1,8)	ta	PN	nsubj	-1	chongyu	VA	ccomp	$+2$	shuo	VV	1	0
(4,6)	muqian	NT	tmod	-2	shang	LC	loc	-2	chongyu	VA	0	0
(6,7)	shang	LC	loc	-2	gongyou	NN	nsubj	-1	chongyu	VA	0	1

(d)

Figure 1: Illustration: (a) neural network classifier architecture with two hidden layers; (b) an aligned Chinese-English parallel sentence pair; and sample extracted training instances and features for (c) head-child classifier and (d) sibling classifier. The label 1 or 0 indicates whether the two words need to be swapped or kept in order, respectively.

  – $+2$ if x_c is on the right of x_h and there is at least one other child between them

- A Boolean $\omega(x_h, x_c)$ to indicate if any punctuation symbol, which is also the child of x_h, exists between x_h and x_c

The sibling classifier predicts the order of the translated words of two source words x_l and x_r, where x_l is to the left of x_r and both have the common head word x_h, using the following features:

- The left child word x_l, its POS tag $T(x_l)$, the dependency label $L(x_l)$ linking x_l to x_h, and the signed distance to its head $d(x_h, x_l)$
- The right child word x_r, its POS tag $T(x_r)$, the dependency label $L(x_r)$ linking x_r to x_h, and the signed distance to its head $d(x_h, x_r)$
- The head word x_h and its POS tag $T(x_h)$
- A Boolean $\omega(x_l, x_r)$ to indicate if any punctuation symbol, which is also the child of x_h, exists between x_l and x_r

Feed-Forward Layers

As shown in Figure 1a, the classifier is a feed-forward neural network whose input layer contains the features. Each feature is mapped by a lookup table to a continuous vector representation, and the resulting vectors are concatenated and fed into (multiplied by) a series of hidden layers (weight matrices) using the rectified linear activation function, $relu(x) = \max(0, x)$. Given the hidden-layer-transformed embedding vector $\mathbf{x}$, a weight vector $\mathbf{W}$, and a bias value b, the prediction output σ is defined as:

$$z = \mathbf{W} \cdot \mathbf{x} + b \quad (1)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

We initialize the hidden layers and the embedding layer for non-word features (POS tags, dependency labels, and Boolean indicators) by a random uniform distribution. For word features x_h, x_c, x_l, and x_r, we initialize their embeddings by the dependency-driven embedding scheme of (Bansal, Gimpel, and Livescu 2014). This scheme is a modified skip-gram model, which given an input word, predicts its context (surrounding words), resulting in a mapping such that words with similar surrounding words have similar continuous vector representations (Mikolov et al. 2013). Similarly, defining the dependency information (i.e., label, head word, and child word) as context produces a mapping such that words with similar head and child words have similar continuous vector representations.

Dependency-driven embedding can be obtained from a dependency-parsed corpus, where each training instance is formulated as (note: x_g is the head of x_h):

$$L(x_h)_{<GL>}\ x_{g<G>}\ x_h\ x_c\ L(x_c)_{<L>}$$

The skip-gram model is trained with a window size of 1 (denoting one context item on the left and one on the right).

Following (Bansal, Gimpel, and Livescu 2014), the items marked by <> subscripts serve as the context, have different continuous vector representations from the words (x_h and x_c), and are filtered out from the embedding vocabulary after training.

Neural Network Training

The training instances for the neural classifiers are obtained from a word-aligned parallel corpus. Two source-side words with head-child or sibling relation are extracted with their corresponding order label, *swapped* or *in order*, depending on the positions of their aligned target-side words. Figure 1 shows the training instances extracted with their corresponding features. For the head-child classifier, the features containing the child information are distinguished based on whether the child is on the left or right of the head.

The NN classifiers are trained using back-propagation to minimize the cross-entropy objective function:

$$L = -\frac{1}{T}\sum_{i=1}^{T} y_i \log \hat{y_i} + (1 - y_i)\log(1 - \hat{y_i}) \qquad (3)$$

where x_i is the i-th training instance, y_i is its corresponding label (1 for *swapped* and 0 for *in order*), and $\hat{y_i}$ is the classifier prediction probability for *swapped*. To prevent model overfitting, we used the dropout strategy (Srivastava et al. 2014) on the input embedding layer.

Reordering in Phrase-Based SMT

We adopt the phrase-based SMT approach, using a beam search decoding algorithm (Koehn 2004a). Each source phrase and one of its possible translations represent an alternative in the search for the translated sentence. While the search produces translation from left to right in the translation output order, it picks the source phrases in any order to enable reordering for a language pair with different word order. The translation output is picked based on a score computed by a log-linear model, comprising the weighted sum of feature function values (Och and Ney 2002).

A phrase-based SMT system typically includes a distance-based reordering penalty (DBR) (Koehn, Och, and Marcu 2003), to discourage long-distance reordering, and phrase-based reordering models (PBRM). The latter comprises the phrase-based lexicalized reordering (PBLR) model (Tillmann 2004; Koehn et al. 2005) and the hierarchical reordering (HR) model (Galley and Manning 2008). These models are the conventional reordering models widely used in phrase-based SMT.

Dependency-Based Decoding Features

Phrase-based decoding can take into account the source dependency parse tree to guide its search. To encourage structural cohesion during translation, we add a dependency distortion penalty (DDP) feature (Cherry 2008) to discourage translation output in which the translated words of a phrase in a source dependency parse subtree are split.

We also incorporate the sparse dependency swap (DS) features of our prior work (Hadiwinoto, Liu, and Ng 2016). The features involve considering a source word x being translated during beam search and each of the as yet untranslated source words x', where x' is the head, the child, or the sibling of x in the source dependency parse tree. x' can be on the left of x in the source sentence, resulting in x and x' being *swapped* in the translation output; or x' can be on the right of x, resulting in x and x' following the same order (*in order*). This principle is used to guide word pair translation ordering through sparse feature templates for head-child word pair and sibling word pair, in which each word x in a word pair is represented by its dependency label ($L(x)$), POS tag ($T(x)$), and their combination.

Specifically, the sparse dependency swap (DS) features of (Hadiwinoto, Liu, and Ng 2016) are based on a feature template for a head word x_h and its child word x_c, their dependency labels and POS tags, whether x_h is on the $p \in \{left, right\}$ of x_c in the source sentence, and the ordering of the pair in the translation output $o \in \{in\_order, swapped\}$:

$$H_{hc}(x_h, x_c, p, o) = \begin{bmatrix} h_{hc}(L(x_h), L(x_c), p, o) \\ h_{hc}(T(x_h), T(x_c), p, o) \\ h_{hc}(L(x_h), T(x_c), p, o) \\ h_{hc}(T(x_h), L(x_c), p, o) \end{bmatrix} \qquad (4)$$

Similarly, there is another feature template for two sibling words, x_l on the left of x_r sharing a common head word, their dependency labels and POS tags, and the ordering of the pair in the translation output $o \in \{in\_order, swapped\}$:

$$H_{sib}(x_l, x_r, o) = \begin{bmatrix} h_{sib}(L(x_l), L(x_r), o) \\ h_{sib}(T(x_l), T(x_r), o) \\ h_{sib}(L(x_l), T(x_r), o) \\ h_{sib}(T(x_l), L(x_r), o) \end{bmatrix} \qquad (5)$$

Incorporating Neural Classifier

We incorporate the neural classifier by defining one decoding feature function for the head-child classifier, and another decoding feature function for the sibling classifier. We also employ model ensemble by training multiple head-child and sibling classifiers, each with a different random seed for hidden layer initialization. Within the log-linear model, the value of each neural classifier feature function is its prediction log-probability. Each feature function is assigned a different weight obtained from tuning on development data.

Experimental Setup

Data Set and Toolkits

We conducted experiments on a phrase-based Chinese-to-English SMT system built using Moses (Koehn et al. 2007). Our parallel training corpora are from LDC, which we divide into older corpora[1] and newer corpora[2]. Due to the dominant

[1] LDC2002E18, LDC2003E14, LDC2004E12, LDC2004T08, LDC2005T06, and LDC2005T10.

[2] LDC2007T23, LDC2008T06, LDC2008T08, LDC2008T18, LDC2009T02, LDC2009T06, LDC2009T15, LDC2010T03, LDC2013T11, LDC2013T16, LDC2014T04, LDC2014T11, LDC2014T15, LDC2014T20, and LDC2014T26.

older corpora, we duplicate the newer corpora of various domains ten times to achieve better domain balance. To reduce the possibility of alignment errors, parallel sentences in the corpora that are longer than 85 words in either Chinese (after word segmentation) or English are discarded. In the end, the final parallel texts consist of about 8.8M sentence pairs, 228M Chinese tokens, and 254M English tokens (a token can be a word or punctuation symbol). We also added two dictionaries[3], having 1.81M Chinese tokens and 2.03M English tokens in total, by concatenating them to our training parallel texts. To train the Chinese word embeddings as described above, we concatenate the Chinese side of our parallel texts with Chinese Gigaword version 5 (LDC2011T13), resulting in 2.08B words in total.

All Chinese sentences in our experiment are first word-segmented using a maximum entropy-based Chinese word segmenter (Low, Ng, and Guo 2005) trained on the Chinese Treebank (CTB) segmentation standard. Then the parallel corpus is word-aligned by GIZA++ (Och and Ney 2003) using IBM Models 1, 3, and 4 (Brown et al. 1993)[4]. For building the phrase table, which follows word alignment, the maximum length of a phrase is set to 7 words for both the source and target sides.

The language model (LM) is a 5-gram model trained on the English side of the FBIS parallel corpus (LDC2003E14) and the monolingual corpus English Gigaword version 4 (LDC2009T13), consisting of 107M sentences and 3.8B tokens altogether. Each individual Gigaword sub-corpus[5] is used to train a separate language model and so is the English side of FBIS. These individual language models are then interpolated to build one single large LM, via perplexity tuning on the development set.

Training the neural reordering classifier involves LDC manually-aligned corpora, from which we extracted 572K head-child pairs and 1M sibling pairs as training instances[6], while retaining 90,233 head-child pairs and 146,112 sibling pairs as held-out tuning instances[7]. The latter is used to pick the best neural network parameters.

Our translation development set is MTC corpus version 1 (LDC2002T01) and version 3 (LDC2004T07). This development set has 1,928 sentence pairs in total, 49K Chinese tokens and 58K English tokens on average across the four reference translations. Weight tuning is done by using the pairwise ranked optimization (PRO) algorithm (Hopkins and May 2011).

We parse the Chinese sentences by the Mate parser, which jointly performs POS tagging and dependency parsing (Bohnet and Nivre 2012), trained on Chinese Treebank (CTB) version 8.0 (LDC2013T21).

Our translation test set consists of the NIST MT evaluation sets from 2002 to 2006, and 2008[8].

Baseline System

Our phrase-based baseline SMT system includes the conventional reordering models, i.e., distance-based reordering penalty (DBR) and phrase-based reordering model (PBRM), both phrase-based lexicalized reordering (PBLR) and hierarchical reordering (HR). We also use the dependency-based reordering features, including the distortion penalty (DDP) feature and the sparse dependency swap (DS) features.

To constrain the decoding process, we set punctuation symbols as reordering constraint across which phrases cannot be reordered, as they form the natural boundaries between different clauses. In addition, a distortion limit is set such that reordering cannot be longer than a certain distance. To pick the translation output, we also use n-best minimum Bayes risk (MBR) decoding (Kumar and Byrne 2004) instead of the default maximum a-posteriori (MAP) decoding.

Neural Reordering

We replaced DS features by our dependency-based neural reordering classifier, in which we set the word vocabulary to the 100,000 most frequent words in our parallel training corpora, replacing other words with a special UNK token, in addition to all POS tags, dependency labels, and Boolean features. We set the embedding dimension size to 100, the lower hidden layer dimension size to 200, and the upper hidden layer dimension size to 100. We trained for 100 epochs, with 128 mini-batches per epoch, and used a dropout rate of 0.5. For model ensemble, we trained 10 classifiers for head-child reordering and 10 for sibling reordering, each of which forming one feature function.

Experimental Results

The translation quality of the system output is measured by case-insensitive BLEU (Papineni et al. 2002), for which the brevity penalty is computed based on the shortest reference (NIST-BLEU)[9]. Statistical significance testing between systems is conducted by bootstrap resampling (Koehn 2004b).

Table 1 shows the experimental results. The distortion limit of all the systems is set to 14. As shown in the table, when the word embedding features are initialized using dependency context (Bansal, Gimpel, and Livescu 2014), which is our default scheme, our translation system with single neural classifier is able to improve over our strong baseline system (DBR+PBRM+DDP+DS) by +0.32 BLEU point, while an ensemble model of 10 neural classifiers improves over our baseline system by +0.57 BLEU point. The results show that the neural reordering classifier is able to replace the sparse dependency swap features and achieves better performance.

In addition to the dependency-driven embedding initialization scheme of (Bansal, Gimpel, and Livescu 2014), we are also interested in testing other word embedding schemes. Additional experiments use two other initialization schemes: (1) random initialization and (2) the original skip-gram model of (Mikolov et al. 2013) with a window size of 5.

[3] LDC2002L27 and LDC2005T34.

[4] The default when running GIZA++ with Moses.

[5] AFP, APW, CNA, LTW, NYT, and Xinhua

[6] LDC2012T20, LDC2012T24, LDC2013T05, LDC2013T23, LDC2014T25, LDC2015T04, and LDC2015T18.

[7] LDC2012T16.

[8] LDC2010T10, LDC2010T11, LDC2010T12, LDC2010T14, LDC2010T17, and LDC2010T21.

[9] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl

Dataset	Baseline				Neural reordering classifier				
	Conventional		Sparse dependency						
	DBR	DBR +PBRM	DDP+DS	DBR+PBRM +DDP+DS	DDP+NR	DBR+PBRM+DDP+NR			
					10s-D	1s-D	10s-R	10s-S	10s-D
Dev	37.44	40.04	40.80	41.39	40.76	40.96	40.18	40.80	41.21
NIST02	37.23	39.19	39.96**†	40.48**††	40.85**††	40.59**††	40.07**††	40.57**††	40.87**††‡
NIST03	37.24	39.44	39.72**	40.88**††	40.75**††	41.11**††	40.15**††	40.67**††	41.46**††‡‡
NIST04	37.66	40.26	40.04**	40.97**††	40.85**††	41.29**††‡	40.65**†	41.00**††	41.70**††‡‡
NIST05	37.32	39.65	39.34**	41.26**††	40.79**††	41.32**††	40.12**	41.37**††	41.56**††
NIST06	36.15	38.70	37.79**	39.15**†	39.25**	39.44**††‡	38.00**	39.46**††	39.95**††‡‡
NIST08	28.47	30.11	29.57**	31.12**††	31.17**††	32.02**††‡‡	29.70**	31.28**††	31.76**††‡‡
Average	35.68	37.89	37.74**	38.98**††	38.94**††	39.30**††	38.12**††	39.06**††	**39.55****††‡‡

Table 1: BLEU scores (%) of our neural reordering (**NR**) approach, either single system (**1s**) or 10-system ensemble (**10s**), with different embedding initialization schemes, i.e., random (**R**), skip-gram (**S**), or dependency-driven (**D**). We used the following prior reordering features as baseline: (1) distance-based reordering (**DBR**) (Koehn, Och, and Marcu 2003); (2) phrase-based reordering models (**PBRM**), comprising phrase-based lexicalized reordering (Tillmann 2004; Koehn et al. 2005) and hierarchical reordering (Galley and Manning 2008); (3) dependency distortion penalty (**DDP**) (Cherry 2008); and (4) sparse dependency swap features (**DS**) (Hadiwinoto, Liu, and Ng 2016). Statistical significance testing compares our approach with **DBR** ($*$: significant at $p < 0.05$; $**$: significant at $p < 0.01$), with **DBR+PBRM** (†: significant at $p < 0.05$; ††: significant at $p < 0.01$), and with **DBR+PBRM+DDP+DS** (‡: significant at $p < 0.05$; ‡‡: significant at $p < 0.01$).

As shown in Table 1, using dependency-driven embedding initialization scheme yields the best improvement over our baseline. On the other hand, random initialization of word embedding yields worse results compared to our baseline, showing a significant drop. Using the skip-gram word embedding model yields average results comparable to the baseline.

We are also interested in testing the performance of the dependency-based reordering features in the absence of the conventional phrase-based reordering models. Table 1 shows that the system with dependency distortion penalty (DDP) and sparse dependency swap (DS) features is unable to outperform the system with only conventional phrase-based reordering models (DBR+PBRM). However, our neural classifier approach, without the conventional reordering models, significantly outperforms the conventional reordering models by +1.05 BLEU point.

Discussion

Dependency swap features capture the dependency label and POS tag of the two words to be reordered, but not the actual words themselves. While using words as sparse features may result in too many parameters, the continuous word representation in our neural approach alleviates this problem. In addition, the neural network model also learns useful combinations of individual features. While dependency swap features (Hadiwinoto, Liu, and Ng 2016) define features as pairs of dependency label and POS tag, the hidden layer of a NN can dynamically choose the information to take into account for the reordering decision.

Using neural classifiers with dependency-based word embedding initialization yields significant improvement, whereas random initialization and skip-gram initialization of word embeddings yield no improvement. This shows the importance of capturing dependency information in the word embeddings for reordering.

Figure 2 shows the baseline phrase-based SMT system with conventional phrase-based reordering models (DBR+PBRM) and sparse dependency swap features produces an incorrect translation output. The sparse dependency swap features prefer the Chinese words "zhiyi (one of)" and "zuzhi (organization)", where "zhiyi" is the head word of "zuzhi", to remain in order after translation, based on their dependency labels and POS tags. However, the Chinese expression "$NOUN$ zhiyi" should be swapped in the English translation, resulting in "one of $NOUN$"[10].

Our experimental results also show that without conventional phrase-based reordering models, the sparse dependency-based features are unable to outperform the conventional reordering models, whereas the neural dependency-based reordering model outperforms the conventional reordering models. This further demonstrates the strength of our dependency-based neural reordering approach.

Our approach applies syntax to SMT with beam search decoding. This is different from prior approaches requiring chart parsing decoding such as the hierarchical phrase-based (Chiang 2007), tree-to-string (Liu, Liu, and Lin 2006), string-to-tree (Marcu et al. 2006), and tree-to-tree (Zhai et al. 2011) SMT approaches.

The end-to-end neural MT (NMT) approach has recently been proposed for MT. However, the most recent NMT papers tested on the same NIST Chinese-to-English test sets (Wang et al. 2016; Zhang et al. 2016) show lower absolute BLEU scores (by 2 to 7 points) compared to our scores. Following the approach of (Junczys-Dowmunt, Dwojak, and

[10] The ordering is further aggravated by wrongly swapping "ISO" and "zuzhi (organization)", due to the translation output score being the weighted sum of features including LM, which prefers such a translation.

Source sentence ISO 是 目前 世界 上 两 大 国际 标准化 组织 之一 。
Reference translation: ISO is one of the world's two international standardization organizations.
DBR+PBRM: The two major International Organization for Standardization (ISO is one of the world.
DBR+PBRM+DDP+DS: The International Organization for Standardization (ISO is one of the two in the world.
DDP+NR, DBR+PBRM+DDP+NR: The ISO is one of the two major international standardization organization in the world.

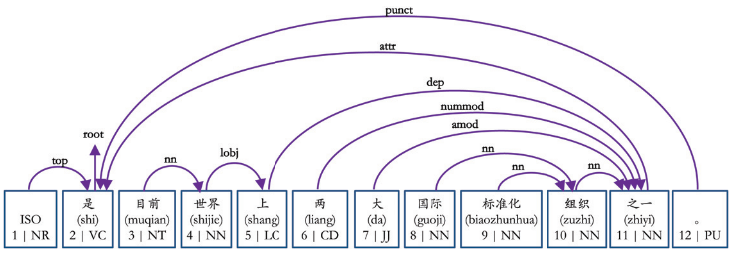


Figure 2: *Left*: a sample sentence and our translation output with distance-based reordering (*DBR*), phrase-based reordering models (*PBRM*), dependency distortion penalty (*DDP*), and sparse dependency swap (*DS*) features, compared to neural reordering in a 10-system ensemble with dependency-driven embeddings (*NR*); *Right*: a source dependency parse tree.

Hoang 2016), our own implemented NMT system (single system without ensemble), when trained on the same corpora and tested on the same NIST test sets in this paper, achieves an average BLEU score of 38.97, lower by 0.58 point compared to our best SMT system ($p < 0.01$). This shows that our neural dependency-based reordering model outperforms the NMT approach. NMT also requires longer time to train (18 days) compared to our best SMT system (3 days).

Related Work

Phrase-based SMT reordering can utilize the dependency parse of the input sentence. Chang et al. (2009) utilized the traversed paths of dependency labels to guide phrase reordering. Hadiwinoto, Liu, and Ng (2016) introduced a technique to determine the order of two translated words with corresponding source words that are related through the dependency parse during beam search. They defined sparse decoding features to encourage or penalize the reordering of two words, based on the POS tag and dependency relation label of each word, but not the words themselves.

Neural reordering models have been applied to re-rank translation candidates generated by the translation decoder. Li et al. (2014) introduced a recursive auto-encoder model to represent phrases and determine the phrase orientation probability. Cui, Wang, and Li (2016) introduced long short-term memory (LSTM) recurrent neural networks to predict the translation word orientation probability. These approaches did not use dependency parse and they were not applied directly during decoding.

Source dependency parse is also used in the pre-ordering approach, which pre-orders words in a source sentence into target word order and then translates the target-ordered source sentence into the target language. While the pre-ordering step typically utilizes a classifier with feature combinations (Lerner and Petrov 2013; Jehl et al. 2014), a neural network can replace the classifier to avoid feature combination. De Gispert, Iglesias, and Byrne (2015) introduced a feed-forward neural network to pre-order the dependency parse tree nodes (words). However, they did not explore dependency-driven embeddings and model ensemble. Miceli-Barone and Attardi (2015) treat pre-ordering as a traversal on the dependency parse tree, guided by a recurrent neural network. In these approaches, the translation possibility is limited to one target ordering. In contrast, applying a neural reordering model jointly with beam search allows for multiple ordering alternatives and interaction with other models, such as the phrase-based reordering models. We can even build multiple neural models (ensemble) and assign a different weight to each of them to optimize translation quality.

Our neural reordering classifier serves as a decoding feature function in SMT, leveraging the decoding. This is similar to prior work on neural decoding features, i.e., neural language model (Vaswani et al. 2013) and neural joint model (Devlin et al. 2014), a source-augmented language model. However, these features are not about word reordering.

While continuous representation of words is originally defined for words (Mikolov et al. 2013), we also define continuous representation for POS tags, dependency labels, and indicator features. Extending continuous representation to non-word features is also done in neural dependency parsing (Chen and Manning 2014; Andor et al. 2016), which shows better performance by using continuous feature representation over the traditional discrete representation.

Conclusion

We have presented a dependency-based reordering approach for phrase-based SMT, guided by neural classifier predictions. It shows that MT can be improved by a neural network approach by not requiring explicit feature combination and by using dependency-driven continuous word representation. Our experiments also show that our neural reordering approach outperforms our prior reordering approach employing sparse dependency-based features.

References

Andor, D.; Alberti, C.; Weiss, D.; Severyn, A.; Presta, A.; Ganchev, K.; Petrov, S.; and Collins, M. 2016. Globally normalized transition-based neural networks. In *ACL 2016*.

Bansal, M.; Gimpel, K.; and Livescu, K. 2014. Tailoring continuous word representation for dependency parsing. In *ACL 2014 Short Papers*.

Bohnet, B., and Nivre, J. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL 2012*.

Brown, P. F.; Della Pietra, V. J.; Della Pietra, S. A.; and Mercer, R. L. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics* 19(2).

Chang, P.-C.; Tseng, H.; Jurafsky, D.; and Manning, C. D. 2009. Discriminative reordering with Chinese grammatical relations features. In *SSST-3*.

Chen, D., and Manning, C. D. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP 2014*.

Cherry, C. 2008. Cohesive phrase-based decoding for statistical machine translation. In *ACL-08: HLT*.

Chiang, D. 2007. Hierarchical phrase-based translation. *Computational Linguistics* 33(2).

Cui, Y.; Wang, S.; and Li, J. 2016. LSTM neural reordering feature for statistical machine translation. In *NAACL HLT 2016*.

de Gispert, A.; Iglesias, G.; and Byrne, B. 2015. Fast and accurate preordering for SMT using neural networks. In *NAACL HLT 2015*.

Devlin, J.; Zbib, R.; Huang, Z.; Lamar, T.; Schwartz, R.; and Makhoul, J. 2014. Fast and robust neural network joint models for statistical machine translation. In *ACL 2014*.

Galley, M., and Manning, C. D. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP 2008*.

Hadiwinoto, C.; Liu, Y.; and Ng, H. T. 2016. To swap or not to swap? Exploiting dependency word pairs for reordering in statistical machine translation. In *AAAI-16*.

Hopkins, M., and May, J. 2011. Tuning as ranking. In *EMNLP 2011*.

Jehl, L.; de Gispert, A.; Hopkins, M.; and Byrne, W. 2014. Source-side preordering for translation using logistic regression and depth-first branch-and-bound search. In *EACL 2014*.

Junczys-Dowmunt, M.; Dwojak, T.; and Hoang, H. 2016. Is neural machine translation ready for deployment? A case study on 30 translation directions. *CoRR* abs/1610.01108.

Koehn, P.; Axelrod, A.; Mayne, A. B.; Callison-Burch, C.; Osborne, M.; and Talbot, D. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *IWSLT 2005*.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E. 2007. Moses: Open source toolkit for statistical machine translation. In *the ACL 2007 Demo and Poster Sessions*.

Koehn, P.; Och, F. J.; and Marcu, D. 2003. Statistical phrase-based translation. In *HLT-NAACL 2003*.

Koehn, P. 2004a. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA 2004*.

Koehn, P. 2004b. Statistical significance tests for machine translation evaluation. In *EMNLP 2004*.

Kumar, S., and Byrne, W. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *HLT-NAACL 2004*.

Lerner, U., and Petrov, S. 2013. Source-side classifier preordering for machine translation. In *EMNLP 2013*.

Li, P.; Liu, Y.; Sun, M.; Izuha, T.; and Zhang, D. 2014. A neural reordering model for phrase-based translation. In *COLING 2014*.

Liu, Y.; Liu, Q.; and Lin, S. 2006. Tree-to-string alignment template for statistical machine translation. In *ACL-COLING 2006*.

Low, J. K.; Ng, H. T.; and Guo, W. 2005. A maximum entropy approach to Chinese word segmentation. In *SIGHAN4*.

Marcu, D.; Wang, W.; Echihabi, A.; and Knight, K. 2006. SPMT: statistical machine translation with syntactified target language phrases. In *EMNLP 2006*.

Miceli-Barone, A. V., and Attardi, G. 2015. Non-projective dependency-based pre-reordering with recurrent neural network for machine translation. In *ACL 2015*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representation in vector space. In *ICLR 2013 Workshop*.

Och, F. J., and Ney, H. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL 2002*.

Och, F. J., and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1).

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL 2002*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(Jun).

Tillmann, C. 2004. A unigram orientation model for statistical machine translation. In *HLT-NAACL 2004: Short Papers*.

Vaswani, A.; Zhao, Y.; Fossum, V.; and Chiang, D. 2013. Decoding with large-scale neural language models improves translation. In *EMNLP 2013*.

Wang, M.; Lu, Z.; Li, H.; and Qun, L. 2016. Memory-enhanced decoder for neural machine translation. In *EMNLP 2016*.

Zhai, F.; Zhang, J.; Zhou, Y.; and Zong, C. 2011. Simple but effective approaches to improving tree-to-tree model. In *MT Summit XIII*.

Zhang, B.; Xiong, D.; Su, J.; Duan, H.; and Zhang, M. 2016. Variational neural machine translation. In *EMNLP 2016*.