A Discriminative Latent Variable-Based "DE" Classifier for Chinese–English SMT

Jinhua Du and **Andy Way**
CNGL, School of Computing
Dublin City University
{jdu, away}@computing.dcu.ie

Abstract

Syntactic reordering on the source-side is an effective way of handling word order differences. The 的 (DE) construction is a flexible and ubiquitous syntactic structure in Chinese which is a major source of error in translation quality. In this paper, we propose a new classifier model — discriminative latent variable model (DPLVM) — to classify the DE construction to improve the accuracy of the classification and hence the translation quality. We also propose a new feature which can automatically learn the reordering rules to a certain extent. The experimental results show that the MT systems using the data reordered by our proposed model outperform the baseline systems by 6.42% and 3.08% relative points in terms of the BLEU score on PB-SMT and hierarchical phrase-based MT respectively. In addition, we analyse the impact of DE annotation on word alignment and on the SMT phrase table.

1 Introduction

Syntactic structure-based reordering has been shown to be significantly helpful for handling word order issues in phrase-based machine translation (PB-SMT) (Xia and McCord, 2004; Collins et al., 2005; Wang et al., 2007; Li et al., 2007; Elming, 2008; Chang et al., 2009). It is well-known that in MT, it is difficult to translate between Chinese–English because of the different word orders (cf. the different orderings of head nouns and relative clauses). Wang et al. (2007) pointed out that Chinese differs from English in several important respects, such as relative clauses appearing before the noun being modified, prepositional phrases often appearing before the head they modify, etc. Chang et al. (2009) argued that many of the structural differences are related to the ubiquitous Chinese structural particle phrase 的 (DE) construction, used for a wide range of noun modification constructions (both single word and clausal) and other uses. They pointed out that DE is a major source of word order error when a Chinese sentence is translated into English due to the different ways that the DE construction can be translated.

In this paper, we focus on improving the classification accuracy of DE constructions in Chinese as well as investigating its impact on translation quality. From the grammatical perspective, the 的(DE) in Chinese represents the meaning of "noun modification" which generally is shown in the form of a Noun phrase (NP) [A DE B]. A includes all the words in the NP before DE and B contains all the words in the NP after DE. Wang et al. (2007) first introduced a reordering of the DE construction based on a set of rules which were generated manually and achieved significant improvements in translation quality. Chang et al. (2009) extended this work by classifying DE into 5 finer-grained categories using a log-linear classifier with rich features in order to achieve higher accuracy both in reordering and in lexical choice. Their experiments showed that a higher

accuracy of the DE classification improved the accuracy of reordering component, and further indirectly improved the translation quality in terms of BLEU (Papineni et al., 2002) scores.

We regard the DE classification as a labeling task, and hence propose a new model to label the DE construction using a discriminative latent variable algorithm (DPLVM) (Morency et al., 2007; Sun and Tsujii, 2009), which uses latent variables to carry additional information that may not be expressed by those original labels and capture more complicated dependencies between DE and its corresponding features. We also propose a new feature defined as "tree-pattern" which can automatically learn the reordering rules rather than using manually generated ones.

The remainder of this paper is organised as follows. In section 2, we introduce the types of word order errors caused by the DE construction. Section 3 describes the closely related work on DE construction. In section 4, we detail our proposed DPLVM algorithm and its adaptation to our task. We also describe the feature templates as well as the proposed new feature used in our model. In section 5, the classification experiments are conducted to compare the proposed classification model with a log-linear model. Section 6 reports comparative experiments conducted on the NIST 2008 data set using two sets of reordered and non-reordered data. Meanwhile, in section 7, an analysis on how the syntactic DE reordering affects word alignment and phrase table is given. Section 8 concludes and gives avenues for future work.

2 The Problem of Chinese DE Construction Translation

Although syntactic reordering is an effective way of significantly improving translation quality, word order is still a major error source between Chinese and English translation. Take examples in Figure 1 as an illustration. The errors of three translation results in Figure 1 are from different MT systems, and many errors relate to incorrect reordering for the 的 (DE) structure.

These three translations are from different Hiero systems. Although Hiero has an inherent reordering capability, none of them correctly reordered "bad reputation" and "middle school" around the DE. Chang et al. (2009) suggested that this is because it is not sufficient to have a formalism which supports phrasal reordering. They claimed it is necessary to have sufficient linguistic modeling, so that the system knows when and how much to rearrange.

Source: 当地(local) 一所(a) 名声不佳(bad reputation) 的(with) 中学(middle school)
Reference: 'a local middle school with a bad reputation'
Team 1: 'a bad reputation of the local secondary school'
Team 2: 'the local a bad reputation secondary school'
Team 3: 'a local stigma secondary schools'

Figure 1: Examples of DE construction translation errors from (Chang et al., 2009)

Figure 2 gives an example illustrating how the reordering of DE construction influences the translation of a Chinese sentence. We can see that if we can properly recognise the DE construction [A DE B] and correctly perform the reordering, we can achieve a closer word order with English and hence a good English translation even it is literal.

Although the Hiero system has a strong reordering capability in its generalised phrases, it still cannot process some complicated and flexible cases of DE construction like those in Figure 1. Therefore, a lot of work has gone into word reordering before decoding so that the Chinese sentences have a closer word order with corresponding English sentences.

3 Related Work on DE Construction

To address the word order problems of the DE construction, Wang et al. (2007) proposed a syntactic reordering approach to deal with structural differences and to reorder source language sentences to be much closer to the order of target language sentences. They presented a set of manually generated syntactic rules to determine whether a 的(DE) construction should be reordered or not before translation, such as "For DNPs consisting of 'XP+DEG', reorder if XP is PP or LCP" etc. (cf. (Wang et al., 2007)). The deficiency of their algorithm is that they did not fully consider the flexibility of the DE construction, as it can be translated in many different ways.

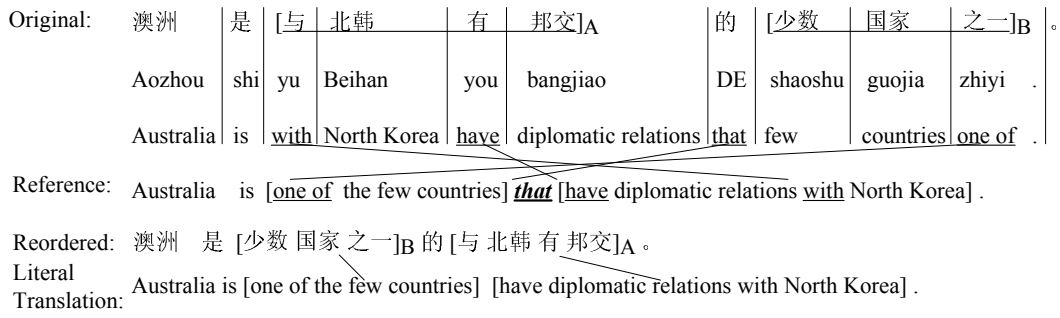

Figure 2: An example of DE construction reordering (extended from the original figure in (Chiang, 2005))

Chang et al. (2009) extended the work of (Wang et al., 2007) and characterised the DE structures into 5 finer-grained classes based on their syntactic behaviour. They argued that one possible reason why the 的(DE) construction remains problematic is that previous work has paid insufficient attention to the many ways that the 的 (DE) construction can be translated, as well as the rich structural cues which exist for these translations.

For a Chinese noun phrase [A 的 B], it can be categorized into one of the following five classes (cf. (Chang et al., 2009) for some real examples of each class):

- A B (label: DE$_{AB}$)

  In this category, A on the Chinese side is translated as a pre-modifier of B. In most cases A is an adjectival form.

- B preposition A (label: DE$_{BprepA}$)

  There are several cases that are translated into the form B preposition A.

- A's B (label: DE$_{AsB}$)

  In this class, the English translation is an explicit s-genitive case. This class occurs much less often but is still interesting because of the difference from the of-genitive.

- relative clause (label: DE$_{relc}$)

  In this class, the relative clause would be introduced by a relative pronoun or be a reduced relative clause.

- A preposition B (label: DE$_{AprepB}$)

  This class is another small one. The English translations that fall into this class usually have some number, percentage or level word in the Chinese A.

Chang et al. (2009) used 6 kinds of features for DE classification, namely part-of-speech tag of DE (DEPOS), Chinese syntactic patterns appearing before DE (A-pattern), unigrams and bigrams of POS tags(POS-ngram), suffix unigram and bigram of word (Lexical), Semantic class of words (SemClass) and Re-occurrence of nouns (Topicality). A conditional log-linear classifier (Chang et al., 2009) is trained to classify each DE based on features extracted from the parsed data.

4 Discriminative Probabilistic Latent Variable Model

4.1 Motivation

Based on the discussion so far, we can see that:

- syntactic reordering of the DE construction in Chinese is an effective way to improve the translation quality;
- classifying the DE construction into finer-grained categories could achieve better reordering and translation performance;
- classification accuracy of the DE construction in Chinese has a significant impact on SMT performance.

Driven by these three points, especially the third one, we propose a DPLVM-based classifier to improve classification accuracy. In natural language

processing (NLP) such as sequential labeling (Sun and Tsujii, 2009), DPLVM demonstrated excellent capability of learning latent dependencies of the specific problems, and have outperformed several commonly-used conventional models, such as support vector machines, conditional random fields and hidden Markov models.

4.2 DPLVM Algorithm

In this section, we theoretically introduce the definition and mathematical description of the DPLVM algorithm used in NLP tasks (Sun and Tsujii, 2009).

Given a sequence of observations $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ and a sequence of labels $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$, the task is to learn a mapping between $\mathbf{x}$ and $\mathbf{y}$. y_i is a class label and is a member of a set $\mathbf{Y}$ of possible class labels. DPLVM also assumes a sequence of latent variables $\mathbf{h} = \{h_1, h_2, \ldots, h_m\}$, which is hidden in the training examples.

The DPLVM is defined as in (1) (Morency et al., 2007; Sun and Tsujii, 2009):

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h}, \mathbf{x}, \Theta) P(\mathbf{h}|\mathbf{x}, \Theta) \quad (1)$$

where Θ are the parameters of the model. It can be seen that the DPLVM equates to a CRF model if it has only one latent variable for each label.

For the sake of efficiency, the model is restricted to have disjoint sets of latent variables associated with each class label. Each h_j is a member in a set $\mathbf{H}_{y_j}$ of possible latent variables for the class label y_j. We define $\mathbf{H}$ as the union of all $\mathbf{H}_{y_j}$ sets, so sequences which have any $h_j \notin \mathbf{H}_{y_j}$ will by definition have $P(\mathbf{y}|\mathbf{x}, \Theta) = 0$, so that the model can be rewritten as in (2):

$$P(\mathbf{y}|\mathbf{x}, \Theta) = \sum_{\mathbf{h} \in \mathbf{H}_{y_1} \times \ldots \mathbf{H}_{y_m}} P(\mathbf{h}|\mathbf{x}, \Theta) \quad (2)$$

where $P(\mathbf{h}|\mathbf{x}, \Theta)$ is defined by the usual conditional random field formulation, as in (3):

$$P(\mathbf{h}|\mathbf{x}, \Theta) = \frac{\exp \Theta \cdot \mathbf{f}(\mathbf{h}, \mathbf{x})}{\sum_{\forall \mathbf{h}} \exp \Theta \cdot \mathbf{f}(\mathbf{h}, \mathbf{x})} \quad (3)$$

in which $\mathbf{f}(\mathbf{h}, \mathbf{x})$ is a feature vector. Given a training set consisting of n labeled sequences (x_i, y_i), for $i = 1 \ldots n$, parameter estimation is performed by optimizing the objective function in (4):

$$L(\Theta) = \sum_{i=1}^{n} \log P(y_i|x_i, \Theta) - R(\Theta) \quad (4)$$

The first term of this equation is the conditional log-likelihood of the training data. The second term is a regularizer that is used for reducing overfitting in parameter estimation.

For decoding in the test stage, given a test sequence $\mathbf{x}$, we want to find the most probable label sequence y^*, as in (5):

$$\mathbf{y}^* = \arg\max_{y} P(\mathbf{y}|\mathbf{x}, \Theta^*) \quad (5)$$

Sun and Tsujii (2009) argued that for latent conditional models like DPLVMs, the best label path $\mathbf{y}^*$ cannot directly be generated by the Viterbi algorithm because of the incorporation of hidden states. They proposed a latent-dynamic inference (LDI) method based on A^* search and dynamic programming to efficiently decode the optimal label sequence $\mathbf{y}^*$. For more details of the LDI algorithm, refer to (Sun and Tsujii, 2009).

In our experiments, we use the open source toolkit of DPLVM[1] and adapt it to our special requirements based on the different features and scenarios.

4.3 Data and DE Annotation

We use the 5 classes of DE of (Chang et al., 2009) shown in Section 3 to label DE using our DPLVM model. In order to fairly compare the classification performance between that of Chang et al. (2009) and our proposed classifiers, we use the same data sets and conditions to train and test the classifier. The data sets are the Chinese Treebank 6.0 (LDC2007T36) and the English–Chinese Translation Treebank 1.0 (LDC2007T02). For more details about the data sets, refer to (Chang et al., 2009). There are 3523 DEs in the data set, with 543 of them in the "other" category which do not belong to any of the 5 pre-defined classes. In the classification experiments, the "other" class is excluded[2] and 2980 DEs remain, each of which

[1] http://www.ibis.t.u-tokyo.ac.jp/XuSun

[2] In the classification experiments of Chang et al. (2009), the "other" class was excluded, so in order to carry out a

is manually annotated with DE labels for the purpose of classifier training and evaluation.

In order to match the training and testing conditions, we used a parser trained on CTB6 excluding files 1-325 to parse the data sets with DE annotation and extract parse-related features rather than using gold-standard parses (same conditions as in (Chang et al., 2009)). It is worth noting that in the Chinese Treebank, there are two types of POS tag for DE in NPs, namely DEC and DEG. However, as a result of using a trained parser, the POS tags of DE might have other values than DEC and DEG. In our data set, there are four other POS tags, namely {AS, DER, DEV,SP}.

4.4 Labels and Features in DPLVM Model

In our task, we use the 5 class labels of DE constructions in NPs, namely DE_{AB}, DE_{AprepB}, DE_{AsB}, DE_{BprepA}, DE_{relc}.

Note that in the case of the DE construction in Chinese, it is different from traditional sequence labeling tasks such as POS tagging, parsing etc. We only need to label one word in the NP structure, i.e. the 的(DE) in a Chinese NP [A DE B]. Therefore the sequence labeling task becomes efficient and speedy using the DPLVM algorithm.

Based on our task, the mathematical conditions for DE classification in a sequence of [A DE B] are denoted as follows:

- **Sequence of Observations:**

  $\mathbf{x} = x_1, \ldots, x_l, x_{DE}, x_k, \ldots, x_m$, where A=$\{x_1, \ldots, x_l\}$, x_{DE} is the Chinese character 的 (DE), and B=$\{x_k, \ldots, x_m\}$;

- **Set of Labels:**

  $\mathbf{Y} = \{y_i | 1 \leq i \leq 5\}$, in which the five labels are DE_{AB}, DE_{AprepB}, DE_{AsB}, DE_{BprepA}, DE_{relc}.

- **Latent Variables:**

  $\mathbf{h} = h_1, h_2, \ldots, h_m$, where $m = 3$ in our task.

We employ five features as well in the DPLVM model, namely DEPOS, POS-gram, lexical features, SemClass as well as a new feature: tree-pattern, which is discussed below.

fair comparison, we did so too. For the SMT experiments, however, we kept it.

We did not add the sixth feature used in (Chang et al., 2009) – topicality – in our classifier because we do not consider it to be a very useful in a data set in which the sentences which are randomly stored. In such a corpus, the content between any adjacent sentences are irrelevant in many cases.

The new feature and the templates of all features used in our task are defined as:

DEPOS:
As mentioned in section 4.3, there are 6 kinds of POS tags of DE. Thus, the feature template is defined as in (5):

$\mathbf{T}_{depos} = \{d_{DE} | d_{DE} \in \textbf{DP}\}$, where **DP** = {AS, DEC, DEG,DER,DEV,SP}. (5)

Tree-pattern:
Chang (2009) used an A-pattern feature which is an indicator function that fires when some syntactic rules are satisfied, such as "A is ADJP if A+DE is a DNP with the form of 'ADJP+DEG'", etc. These rules are induced manually based on the grammatical phenomena at hand. Here we propose a more generalised feature defined as "tree-pattern" to automatically learn the reordering from the training data.

We consider all the sub-tree structures around DE without any word POS tags. For example, consider the parse structure (an example in (Chang et al., 2009)) in (6):

(NP (NP (NR 韩国)) (CP (IP (VP (ADVP (AD 最)) (VP (VA 大)))) (DEC 的)) (NP (NN 投资) (NN 对象国)))))) (6)

where the tree-pattern is "NP NP CP IP VP ADVP VP DEC NP". We do not use the word POS tag (except DE) in this feature, such as NR, AD, VA, etc. The intention of this feature is to enable the classifier to automatically learn the structural rules around DE. Given that the position of DE in the parsing of [A DE B] is i, then the feature template is defined as in (7):

$$\mathbf{T}_{tree\_u} = \{t_{i-l}, \ldots, t_{i-1}, t_i, t_{i+1}, \ldots, t_{i+m}\} \quad (7)$$
$$\mathbf{T}_{tree\_b} = \{t_{i-l}t_{i-l+1}, \ldots, t_{i-1}t_i, t_it_{i+1}, \ldots, t_{i+m-1}t_{i+m}\}$$

where $\mathbf{T}_{tree\_u}$ is the sequence of unigrams in connection with DE and $\mathbf{T}_{tree\_b}$ is the sequence of bigrams related to DE; l and m are the window

sizes of A and B respectively. Generally, we use all the unigrams and bigrams in the parsing of A and B in our experiments. We argue that the important advantage of this feature is that it does not depend on manually generated rules, but instead of learns and generalises the reordering rules from the training data directly.

POS-gram:
The POS-ngram feature adds all unigrams and bigrams in A and B. Given that the position of DE is i in [A DE B], the feature template is defined as in (8):

$$\mathbf{T}_{pos\_u} = \{p_{i-l}, \ldots, p_{i-1}, p_{i+1}, \ldots, p_{i+m}\}$$
$$\mathbf{T}_{pos\_b} = \{p_{i-l}p_{i-l+1}, \ldots, \mathbf{p}_{i-1}\mathbf{p}_{i+1}, \ldots, p_{i+m-1}p_{i+m}\} \quad (8)$$

where $\mathbf{T}_{pos\_u}$ and $\mathbf{T}_{pos\_b}$ are uigrams and bigrams in A and B. In the unigrams, we exclude the POS of DE; in the bigrams, we include a bigram pair across DE.

Some other features such as lexical features, SemClass (cf. (Chang et al., 2009) for details) can be defined using similar feature template.

5 Experiments on DPLVM DE Classifier

In this section, we compare the performance of DE classifiers between the DPLVM and log-linear methods.

The accuracy of classification is defined as in (9):

$$\frac{\text{number of correctly labeled DEs}}{\text{number of all DEs}} \times 100 \quad (9)$$

Phrase Type	Log-linear		DPLVM	
	5-A	2-A	5-A	2-A
DEPOS	54.8	71.0	**56.2**	**72.3**
+A-pattern	67.9	83.7	-	-
+Tree-pattern	-	-	**69.6**	**85.2**
+POS-gram	72.1	84.9	**73.6**	**86.5**
+Lexical	74.9	86.5	**76.4**	**87.9**
+SemClass	75.1	86.7	**76.8**	**88.3**
+Topicality	75.4	86.9	-	-

Table 1: Comparison between the two classifiers on 5-class and 2-class accuracy

Table 1 shows the comparison of accuracy, where "5-A" and "2-A" represent the accuracy of the 5-class and 2-class respectively. The 2-class is the categorised classes of DE in (Wang et al., 2007) which are defined as "reordered" and "non-reordered" categories. It can be seen that our DPLVM classifier outperforms the log-linear classifier by 1.4 absolute (1.86% and 1.61% relative respectively) points both on 5-class and 2-class classifications. Furthermore, we see that the DPLVM achieves significantly better performance than the log-linear model only with the simple feature of "DEPOS". As to the new feature "tree-pattern", we can see that it achieves the improvement of 1.5% compared to the "A-pattern" in terms of the accuracy of "2-A". This improvement attributes to the good learning ability of DPLVM as well as the strong generalisation capability of the tree-pattern feature.

In terms of speed, in our task we only need to label the Chinese character DE in the NP structure [A DE B] rather than label the whole sentence, so that we have a feature matrix of $n \times 1$ for each DE. Accordingly, the DPLVM classifier can run efficiently with low memory usage.

6 Experiments on SMT

6.1 Experimental Setting

For our SMT experiments, we used two systems, namely Moses (Koehn et al., 2007) and Moses-chart. The former is the state-of-the-art PB-SMT system while the latter is a new extended system of the Moses toolkit re-implementing the hierarchical PB-SMT (HPB) model (Chiang, 2005). The alignment is carried out by GIZA++ (Och and Ney, 2003) and then we symmetrized the word alignment using the grow-diag-final heuristic. Parameter tuning is performed using Minimum Error Rate Training (Och, 2003).

The training data contains 2,159,232 sentence pairs.The 5-gram language model is trained on the English part of the parallel training data. The development set (devset) is the NIST MT2006 test set and the test set is the NIST MT2008 "current" test set. All the results are reported in terms of BLEU (Papineni et al., 2002) and METEOR (MTR) (Banerjee and Lavie, 2005) scores.

To run the DE classifiers, we use the Stanford Chinese parser (Levy and Manning, 2003) to parse the Chinese side of the MT training data, the

devset and test set.

6.2 Statistics of 5-class DE Annotation

For the DE-annotated MT experiments, after we parse the training data, the devset and the test set, we separately use the two DE classifiers to annotate the DE constructions in NPs in all of the parsed data. Once the DE data are labeled, we pre-process the Chinese data by reordering the sentences only with 的$_{BprepA}$ and 的$_{relc}$ annotations. Table 2 lists the statistics of the DE classes in the MT training data, devset and test set using our DPLVM classifier. "的$_{non}$" denotes the unlabeled 的(DE) which does not belong to any of the 5 classes.

6.3 Experimental Results

The experimental results from the PB-SMT and HPB systems separately using the DPLVM and log-linear classifiers are shown in Table 3.

	PB-SMT			Moses-chart		
	BL	LL	LV	BL	LL	LV
BLEU	22.42	23.47	**23.86**	24.36	24.75	**25.11**
MTR	52.03	53.25	**53.78**	53.37	53.75	**54.21**

Table 3: Experimental results on PB-SMT and Moses-chart. "BL" are the baselines; "LL" indicates the log-linear model-based system; "LV" is our DPLVM method.

The baseline systems indicate that the data is neither categorised into DE classes nor reordered on the Chinese side. We can see that (1) the "LV" method outperformed the "BL" and "LL" by 1.44 absolute (6.42% relative), 0.39 absolute (1.66% relative) BLEU points for PB-SMT, and by 0.75 absolute (3.08% relative), 0.36 absolute (1.45% relative) BLEU points for Moses-chart; (2) the "LV" method achieved the improvements for PB-SMT and Moses-chart in terms of MTR scores compared to the "BL" and "LL" systems. Therefore, using DE classification and reordering on the source-side is helpful in improving translation quality; (3) the results using DPLVM achieve better translation quality than that of the "LL" processed data in terms of BLEU and METEOR (Banerjee and Lavie, 2005) scores, which indirectly shows that DPLVM outperforms the log-linear classification model; and (4) the improvements on both PB-SMT and Moses-chart show that the effectiveness of DE reordering is consistent for different types of MT systems. The results are verified by significance test on 95% confidence interval (Zhang and Vogel, 2004).[3]

7 Analysis

In this section, we plan to evaluate how DE reordering contributes to the improvement of translation quality in two respects, namely word alignment and phrase table.

7.1 Evaluating the Word Alignment

We create a word alignment test set which includes 500 sentences with human alignment annotation, and then add this test set into the MT training corpus. Accordingly, the DE-reordered test set is added into the reordered training corpus as well. Thus, we run GIZA++ using the same configurations for these two sets of data and symmetrize the bidirectional word alignment using grow-diag heuristic. The word alignment of the test set is evaluated with the human annotation using Precision, Recall, F1 and AER measures. The results are reported in Table 4.

	P	R	F1	AER
non-reordered	71.67	62.02	66.49	33.44
reordered	74.02	62.79	67.95	31.98
Gain	**2.35**	**0.77**	**1.46**	**-1.46**

Table 4: Comparison of Precision, Recall, F1 and AER scores of evaluating word alignment on original and reordered data

We can see that in terms of the four measures, the word alignment produced by the reordered data is slightly better than that of the original data. In some sense, we might say that the DE reordering is helpful in improving the word alignment of the training data.

7.2 Evaluating the Phrase Table

Wang et al. (2007) proposed one way to indirectly evaluate the phrase table by giving the same type of input to the baseline and reordered systems,

[3] http://projectile.sv.cmu.edu/research/public/tools/bootStrap/tutorial.htm.

	training		devset		testset	
DE-class	count	percent (%)	count	percent (%)	count	percent (%)
的$_{AB}$	312,679	23.08	523	25.80	453	28.78
的$_{AprepB}$	6,975	0.51	9	0.44	7	0.44
的$_{AsB}$	13,205	0.97	23	1.13	14	0.89
的$_{BprepA}$	658,589	47.31	956	48.05	688	43.71
的$_{relc}$	316,772	23.38	419	20.67	341	21.66
的$_{non}$	46,547	3.44	97	4.79	71	4.51
Total 的	1,354,767	100	2027	100	1574	100

Table 2: The number of different DE classes labeled for training data, devset and testset using the DPLVM classifier

with the consideration that if the reordered system learned a better phrase table, then it may outperform the baseline on non-reordered inputs despite the mismatch and vice versa. However, they did not settle the question as to whether the reordered system can learn better phrase tables.

We also try to use the idea of Wang et al (2007) to carry out the phrase table evaluation on PB-SMT,[4] i.e. we tune the baseline on a reordered devset and then evaluate on a reordered test set; tune the reordered system on a non-reordered devset and then evaluate on a non-reordered test set. The results are shown in Table 5.

Testset	baseline	reordered	
		LL	DPLVM
non-reordered set	22.42	22.76	22.85
reordered set	23.36	23.47	23.86

Table 5: Comparison of BLEU scores in matched and mismatched conditions on PB-SMT.

We find that (1) given the non-reordered test set, the DE reordered system performs better than the baseline system, which is consistent when different DE classifiers are applied; (2) given the reordered test set system, the reordered set produces a better result than the baseline, which is also consistent when different DE classifiers are applied; and (3) the results from the DPLVM-based reordered data are better than those from the LL-based reordered data. From the comparison, one might say that the reordered system was learned a better phrase table and the reordered test set addresses the problem of word order.

To sum up, from the SMT results and the evaluation results on the word alignment and the phrase table, we can conclude that the DE reordering methods contribute significantly to the improvements in translation quality, and it also implies that using DE reordered data can achieve better word alignment and phrase tables.

8 Conclusions and Future Work

In this paper, we presented a new classifier: a DPLVM model to classify the Chinese 的(DE) constructions in NPs into 5 classes. We also proposed a new and effective feature – tree-pattern – to automatically learn the reordering rules using the DPLVM algorithm. The experimental results showed that our DPLVM classifier outperformed the log-linear model in terms of both the classification accuracy and MT translation quality. In addition, the evaluation of the experimental results in section 7 indicates that the DE-reordering approach is helpful in improving the accuracy of the word alignment, and can also produce better phrase pairs and thus generate better translations.

As for future work, firstly we plan to examine and classify the DE constructions in other syntactic structures such as VP, LCP etc. Secondly, we plan to apply the DE-annotated approach in a syntax-based MT system (Zollmann and Venugopal, 2006) and examine the effects. We also intend to improve the classification accuracy of the DE classifier with richer features to further improve translation quality.

[4]The phrases in HPB systems are different from those in PB-SMT because they are variable-based, so we evaluate the hierarchical phrases in (Du and Way, 2010)

Acknowledgment

Many thanks to Dr. Pi-Chuan Chang for providing the source code of her DE classifier and manually DE-annotated training data as well as valuable instruction in their use. Thanks also to Dr. Xu Sun for the source code of his Latent Variable classifier together with help in their use. This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, ACL-2005*, pages 65–72.

Pi-Chuan Chang, Dan Jurafsky and Christopher D. Manning. 2009 Disambiguating "DE" for Chinese-English machine translation. In *Proceedings of the Fourth Workshop on SMT*, pages 215–223.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL'05*, pages 263–270.

Michael Collins, Philipp Koehn, and Ivona Kucerova. newblock 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL05*, pages 531–540.

Jinhua Du and Andy Way. 2010. The impact of source-side syntactic reordering on hierarchical phrase-based SMT. In *Proceedings of the 14th Annual conference of the European Association for Machine Translation*, Saint-Raphael, France.

Jakob Elming. 2008. Syntactic reordering integrated with phrase-based SMT. In *Proceedings of ACL-08 SSST-2*, pages 46–54.

Philipp Koehn, Hieu Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, Wade Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *demonstration session of ACL'07*, pages 177–180.

Roger Levy and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of ACL'03*, pages 439–446.

Chi-Ho Li, Dongdong Zhang, Mu Li, Ming Zhou, Minghui Li and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *proceedings of the ACL'07*, pages 720–727.

Louis-Philippe Morency, Ariadna Quattoni and Trevor Darrell. 2007. Latent-dynamic Discriminative Models for Continuous Gesture Recognition. In *proceedings of CVPR'07*, pages 1–8.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL'03*, pages 160–167.

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

Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the ACL-02*, pages 311–318.

Xu Sun and Jun'ichi Tsujii. 2009. Sequential Labeling with Latent Variables: An Exact Inference Algorithm and An Efficient Approximation. In *Proceedings of The European Chapter of the Association for Computational Linguistics (EACL'09)*, pages 772-780.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of EMNLP-CoNLL*, pages 737–745.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of Coling 2004*, pages 508–514.

Ying Zhang and Stephan Vogel. 2004. Measuring Confidence Intervals for the Machine Translation Evaluation Metrics. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 85–94.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of HLT-NAACL 2006: Proceedings of the Workshop on Statistical Machine Translation*, New York, pages 138–141.