Exploring Diverse Features for Statistical Machine Translation Model Pruning

Mei Tu, Yu Zhou, and Chengqing Zong, *Senior Member, IEEE*

***Abstract*—In phrase-based and hierarchical phrase-based statistical machine translation systems, translation performance depends heavily on the size and quality of the translation table. To meet the requirements of making a real-time response, some research has been performed to filter the translation table. However, most existing methods are always based on one or two constraints that act as hard rules, such as not allowing phrase-pairs with low translation probabilities. These approaches sometimes make constraints rigid because they consider only a single factor instead of composite factors. Based on the considerations above, in this paper, we propose a machine learning-based framework that integrates multiple features for translation model pruning. Experimental results show that our framework is effective by pruning 80% of the phrase-pairs and 70% of the hierarchical rules, while retaining the quality of the translation models when using the BLEU evaluation metric. Our study further shows that our method can select the most useful phrase-pairs and rules, including those that are low in frequency but still very useful.**

***Index Terms*—Classification, statistical machine translation (SMT), syntactic constraints, translation model pruning.**

I. Introduction

RECENTLY, intelligent terminals (such as mobile devices, PDAs and smartphones) have become increasingly popular. Most of these devices would benefit from a real-time translation tool to break the barrier of different languages. However, it is still difficult to produce a real-time as well as acceptable response, because of the limitations of the memory and CPU performance. The memory cost arises when the translation table become large. Therefore, this paper focuses on the problem of pruning the translation model for a statistical machine translation (SMT) system. In SMT, phrase-based translation systems (PBTSs), which include the hierarchical phrase-based translation systems (HPBTSs) [1], are the most popular and mature systems. The main task of a machine translation decoder (such as Moses [2][1]) is to choose the most promising translation candidates from a large-scale translation table with several conditional probabilities. The translation pairs are extracted by a heuristic method [3] based on the word alignments learned from parallel corpora [4]. Thus, the size and quality of the translation table determines the overall translation performance to a large extent.

Generally, two typical sorts of phrase-pairs cause the phrase table to be redundant and much larger than expected. One is that a distinct source phrase corresponds to many translation options (1-to-many). In fact, many options are poorly translated and are never considered in decoding due to limited beam size, which can be discarded safely. The other case is that a distinct source phrase has only one or two translation options (1-to-few). In this case, many phrase-pairs are extracted by accident because of wrong alignments; thus, the source and target sides are not significantly associated. As far as we know, it is difficult to prune them both at a time effectively. For example, some pruning methods use heuristic rules, such as histogram pruning, to cut off overloaded translation options, but those methods always do not work for the second case. While for the second case, some methods adopt Fisher's significance test [7], [8], [17] to prune the weakly associated pairs. However, they may discard many useful pairs unexpectedly, such as named entities that occur rarely in a parallel training corpus, which could damage the translation quality. Other methods using relative entropy [33], [34] are good at pruning redundant phrases, especially for those long phrases that can be replaced by combining shorter phrases. But the score of combined phrases often differs from the original one. As a result, with more and more discarding, some pruned long phrases cannot be reconstructed by shorter ones after score ranking, leading to weakening of the translation performance. Therefore, there is still much room for improvement in achieving a significant pruning rate while maintaining the translation quality. For this aim, we should seek an integration framework to perform the pruning task by accounting for most of the useful factors and measurements.

From the analysis above, in this paper, a simple but effective machine learning-based method is proposed to select the useful translation pairs in the phrase-based translation model (PBTM) or rules in the hierarchical phrase-based translation model (HPBTM). In our method, we consider the pruning task as a classification problem. Along with that approach, many effective heuristic measures mentioned above are encoded as strong representative features under the classification framework, which makes the heuristic measures more flexible than when used as hard rules.

Manuscript received September 03, 2012; revised June 28, 2013; accepted July 04, 2015. Date of publication July 14, 2015; date of current version July 22, 2015. This work was supported in part by the Natural Science Foundation of China under Grants 61333018 and 61403379 and in part by the International Science and Technology Cooperation Program of China under Grant 2014DFA11350. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mei-Yuh Hwang.

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: mtu@nlpr.ia.ac.cn; yzhou@nlpr.ia.ac.cn; cqzong@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2015.2456413

[1] http://www.statmt.org/moses/index.php?n=Main.HomePage

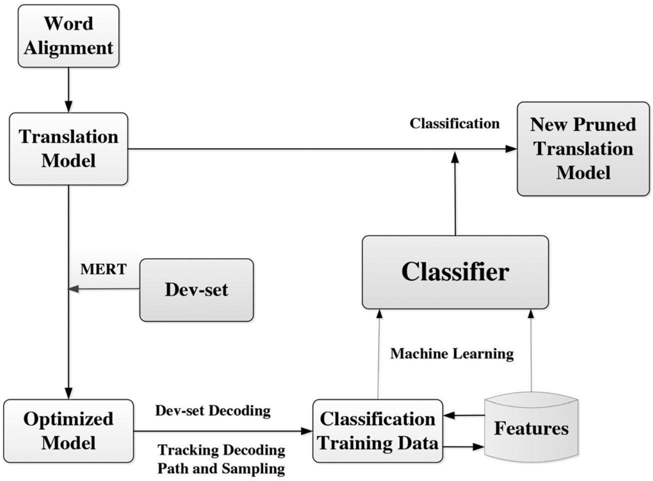


Fig. 1. Pruning process in our classification framework.

Although the concept of using a classifier to improve the BLEU score is not new, the main contribution of this paper is to explore rich statistical and syntactic features that are applied to the classification framework to compact the translation tables of PBTM and HPBTM as much as possible. Experimental results show that our method is effective because it prunes 80% of the phrase-pairs and 70% of the hierarchical rules while retaining the quality of the translation models under the BLEU evaluation metric.

The remainder of this paper is organized as follows: Section II presents the machine learning-based pruning framework. Section III describes how to express the heuristic measures as features and how to utilize those features for our classifier. Our experimental results are presented in Section IV and further analysis is given in Section V. Section VI introduces the related work. Finally, we give concluding remarks and mention future work in Section VII.

II. Overview of Our Proposed Machine Learning-Based Pruning Framework

As mentioned above, our basic idea is to distinguish which phrase-pairs or rules of the whole translation model are potentially more useful than others, and we treat the filtration of the phrase-table and rule-table as a classification problem.

Fig. 1 illustrates the whole classification framework of how we build the classifier and how to make it work for obtaining a pruned translation-table. Given an original translation model, we tune all of the parameters based on a development set using the Minimum Error Rate Training (MERT) process. Then, we trace the decoding path on the development set with the optimized translation model and sample training data for the classifier. Additionally, we extract features and use them to train the classifier based on the training data. Finally, the classifier is utilized to prune the original translation table and thus obtain the final filtered translation table.

III. Details of Our Pruning Method

In this section, we provide details about building a classifier for pruning the translation table.

A. *Training Set for the Classifier*

A key issue in building a classifier is how to obtain the classification training data automatically. Because all of the parameters in a translation model are tuned using the development data (Dev-data in Fig. 1), the tuned translation model can tell which translations are best for the development of source phrases. Thus, it is very natural to extract the classification training data from the tuned decoding path of the development source sentences.

We describe the details of how to obtain positive and negative training data for our classifier as follows. First, we filter out a development translation table (*Dev-TT*) from the whole translation table based on the development source sentences. Then, we trace the search graph during decoding the development set with the optimized parameters, and record every phrase-pair or rule that is in the decoding lattice. Because there is no classification criterion that discriminates good and bad translation pairs, we learn it from data. In this paper, we divide all the phrase-pairs or rules of the *Dev-TT* into three categories, including "actually finally used", "considered but not finally used" and "not considered", as inspired by [5]. Among *Dev-TT*, all of the "actually finally used" phrase-pairs or rules in the final translation are adopted as positive training data, and the "not considered" are adopted as negative training data. The related definitions are given as follows:

$$W = \{Dev - TT\}$$

The development translation table (*Dev-TT*) is filtered out from the whole translation table based on the development data.

$$\{C = TT - In - Beam\}$$

The translation table in beam (*TT-In-Beam*) contains phrase-pairs or rules that have been involved in the translation lattice at least once by the beam search during decoding on the development data.

$$P = \{Used - TT\}$$

The phrase-pairs that are actually used in final translation. We assign P by deriving the search path of the best translation. Note that translation candidates that are identical to the best translation may exist by accident, but we only care the phrase-pairs in the best path. P is a subset of C.

$$N = \{TT - Out - Of - Beam\}$$

The $\{$*TT-Out-Of-Beam*$\}$ is the complement of C in W, consisting of those phrases that were always dropped out of the beam.

Fig. 2 shows the relationship of the positive and negative data. Here, P is the positive data, and N is the negative data. The closed interval C-P (C excludes P) represents the data that are only considered in the decoding but are not chosen for the final translation.

B. *Features for Pruning the Phrase-Based Models*

The features are designed for different translation models. In the following approach, we introduce features for the phrase-based model in detail. Some of the features are inspired by previous studies [18], [19], [21]–[26].

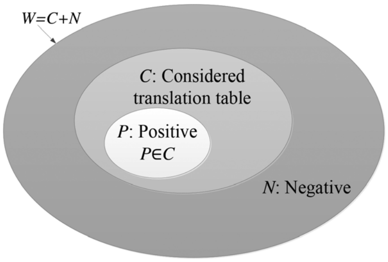


Fig. 2. The relationship between positive data and negative data.

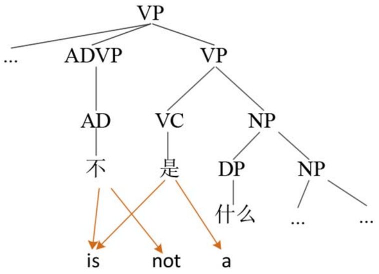


Fig. 3. The source syntactic tree and the alignment of the phrase pair "不 是 什么 ||| is not a". The last word of the source side "什么" is not aligned with any word on the target side.

In general, we use four types of features. The first type is bi-directional translation probabilities, and the second type is syntactic constraints. We also consider the significant value of Fisher's test [7] and the length ratio.

In terms of the translation probabilities, we take two types of probabilities into account. One type is the bi-directional phrase translation probabilities and bi-directional lexicalized translation probabilities, which can be obtained from the translation table. The second type is the reordering probabilities because they might impact whether the phrase-pair should be kept.

For the syntactic constraints, if a span corresponds to a single sub-tree in a syntactic parse tree, it is called a "syntactic phrase", and this definition is naturally to be a syntactic constraint for phrase pruning. The study in [10] used a less strict syntactic constraint. If the source side of a phrase-pair is an illegal syntactic phrase and the first or last word in the source side is not aligned, then the pair will be discarded. However, we think that this rule is still somewhat rigid. Take the following example, "不 是 什么 ||| is not a," where the last word "什么(what)" of the source phrase is unaligned. The syntactic tree of the source phrase is given in Fig. 3.

According to the syntactic constraints used in [10], this phrase-pair should be eliminated, and the phrase-pairs "之 鹰 ||| eagle" and "便捷 的 ||| convenient " should be pruned out for the same reason. However, these pairs could be good translation pairs when decoding. Therefore, we treat the syntactic constraints as features, which enables us to generate a more flexible judgment.

We consider the alignment of the boundary words and the syntax information to be binary-valued features. The source syntactic constraint features are listed as follows:

- *SHS*: if the source phrase starts with a syntactic sub-phrase (more than one word), then $SHS = 1$; otherwise, $SHS = 0$;

TABLE I
BINARY VALUES FOR THE PHRASE PAIR" 不 是 什么 ||| IS NOT A"

Phrase Pair	*不是什么 ||| is not a*
SHS	0
STS	0
SHA	1
STA	0
SSW	0

- *STS*: if the source phrase ends with a syntactic sub-phrase (more than one word), then $STS = 1$; otherwise, $STS = 0$;
- *SHA*: if the first word of the source phrase is aligned, then $SHA = 1$; otherwise, $SHA = 0$;
- *STA*: if the last word of the source phrase is aligned, then $STA = 1$; otherwise, $STA = 0$;
- *SSW*: if the source phrase is a single word, then $SSW = 1$; otherwise, $SSW = 0$;

With the source syntactic feature defined above, the values of the syntactic features of the phrase-pair "不 是 什么 ||| is not a" are given in Table I.

We also consider the target syntactic constraints, which are the same as those of the source syntactic constraints.

With respect to the significant value, previous studies have suggested that Fisher's test is an effective way to prune out most of the phrase-pairs, and a similar result has been verified in our experiments. Therefore, we employ the p-value as a feature in our classifier. Next, we briefly review how to compute the p-value. Given training set that constitutes N parallel sentences and a phrase-pair (s,t) to be valued, we collect $C(s)$, $C(t)$ and $C(s,t)$, which represent the counts of sentences that contain s, t and the pair (s,t), respectively. The probability that the pair (s,t) occurs $C(s,t)$ times by chance is given by the hypergeometric distribution, as follows:

$$P_h(C(s,t)) = \frac{\binom{C(s)}{C(s,t)}\binom{N-C(s)}{C(t)-C(s,t)}}{\binom{N}{C(t)}} \tag{1}$$

The corresponding p-value is the sum of $P_h(C(s',t'))$ under the circumstance that $C(s',t')$ is not less than $C(s,t)$, while $C(s')$ equals $C(s)$ and $C(t')$ equals $C(t)$,

$$Pv(C(s,t)) = \sum_{c=C(s,t)}^{\infty} P_h(c) \tag{2}$$

In addition, we add the length ratio as a feature. Let L_s be the length (the word counts) of the source phrase, and let L_t be the length of the target phrase. Then, we obtain

$$LenRatio = \min(L_s, L_t)/\max(L_s, L_t) \tag{3}$$

C. Features for Pruning the Hierarchical Phrase-Based Models

Because the hierarchical model includes not only the conventional phrase-pairs but also the rules with non-terminals, the hierarchical model becomes more complicated than the common phrase model, in general. Moreover, it is usually more difficult to judge whether a rule with non-terminal is useful compared to a phrase-pair, because the rule with non-terminal has less lexicalized information. Thus, the features used for the phrase-based model cannot be directly applied to the hierarchical model. We must make some modifications.

As mentioned above, the hierarchical phrase-based models include two forms of pairs: those with and those without non-terminals. Features for the rules without non-terminals are the same as those in the phrase-based model; as a result, we do not describe the details here again. In terms of the features for the rules with non-terminals, there are a total of three types.

First, the rule translation probabilities and lexicalized translation probabilities are also accounted for. Second, the LenRatio is slightly different from Equation (3).

$$LenRatio = \min(L_{s\_ter\min al}, L_{t\_ter\min al}) / \max \\ \times (L_{s\_ter\min al}, L_{t\_termianl}) \tag{4}$$

where $L_{s\_ter\min al}$ is the terminal length (word counts) in the left-hand side of a rule and $L_{t\_ter\min al}$ is the terminal length of the right-hand side.

Next, we employ the dependency syntactic features to make a soft constraint on the rules.

The same concept of the *Relax-Well-Formed* structure in [19] is used to compute the dependency features. We first review how the *Relax-Well-Formed* structure is defined. Let $S = w_1 w_2 \ldots w_n$ represent a sentence, and let $d_1 d_1 \ldots d_n$ be the position of the parent word for each word. If w_i is a root, then $d_i = -1$. Given a dependency structure $w_i \ldots w_j$, it will be called a *Relax-Well-Formed* structure if and only if it satisfies the following conditions:

- $d_h \notin [i, j]$Z, where $h \notin [i, j]$
- $\forall k \in [i, j], d_k \in [i, j] \quad or \quad d_k = h$, where $h \notin [i, j]$

Rules that are not *Relax-Well-Formed* are pruned in [19]. In this paper, they are not immediately discarded when the rules are against the *Relax-Well-Formed* structure; instead, we prefer to use a vector of probabilities that indicate the probability that a rule will be kept or discarded under the dependency constraint. We use a triple vector $(\mathrm{D}_l, \mathrm{D}_r, \mathrm{D}_b)$ to represent the probabilities, where D_l is the dependency feature value for the source side of a rule, D_r for the target side, and D_b for both sides. We say that $RWF(\text{rule}) = \text{true}$ if a rule is *Relax-Well-Formed*. Given a rule $(\mathrm{r}_l, \mathrm{r}_r)$, where r_l represents the left side of the rule and r_r represents the right side, we obtain the values in the triple vector for r_l and r_r as follows,

$$D_l = \frac{\sum\limits_{i \in S_s} \delta_{RWF(r_{li})}}{\sum\limits_{i \in S_s} Count(r_{li})} \tag{5}$$

$$D_r = \frac{\sum\limits_{j \in S_t} \delta_{RWF(r_{rj})}}{\sum\limits_{j \in S_t} Count(r_{rj})} \tag{6}$$

$$D_b = \frac{\sum\limits_{i \in S_s} \sum\limits_{j \in S_t} \delta_{RWF(r_{li})} \delta_{RWF(r_{rj})}}{\sum\limits_{i \in S_s} \sum\limits_{j \in S_t} Count(r_{li}, r_{rj})} \tag{7}$$

where S_s or S_t is the set of source or target sentences in the training corpus, respectively. When extracted from the ith sentence, r_l is specified to be r_{li} and has the same meaning as r_{rj}. δ_* is an Kronecker delta function whose value is 1 once $*$ is true; otherwise, its value is 0. $Count(*)$ represents the counts of $*$.

We use this type of dependency syntactic constraints on rules that have non-terminals in the hierarchical phrase-based model instead of the original syntactic constraints in the phrase-based model due to the characteristics of the dependency syntax. Because the non-terminals in the rules match any possible consecutive words in a sentence and cover one or more nodes in a syntactic tree, it is difficult to judge whether a rule with a non-terminal is a legal syntactic rule. Thus, the dependency tree constraints appear to be a good choice because we only need to consider the lexicalized information. At the same time, the dependency constraints can be applied on both the phrase-pairs and the rules; thus, these constraints are more convenient to us when acquiring features in a unified feature-representing model.

IV. Experiments

In our experiments, we focus on the pruning rate of the translation model while considering the translation quality. To compare our methods with other existing methods, a series of experiments have been performed on the Chinese-to-English translation task.

A. Experimental Setup

We use the FBIS corpus as training data, which contains approximately 7.1 million Chinese words and 9.2 million English words. To obtain sufficient training samples for classification, we use the NIST03 and NIST04 corpus as the development set, which contain 919 and 1,788 sentences, respectively. NIST05 and NIST06 are used as test sets, with 1,083 and 1,664 sentences included, respectively. Note that NIST05 and NIST06 are only used as testing data in the translation task and never used for training the classifier.

To obtain the original phrase-based and hierarchical phrase-based translation model, we train a 5-gram language model with SRILM[2] on the FBIS English part. We obtain the source-to-target and target-to-source word alignments by GIZA++[3]. These alignments are then symmetrized with grow-diag-final-and strategy. The translation model is generated by Moses (2010-8-13 Version), using the default parameter settings. For the phrase-based translation model, the maximum length of the phrases in the phrase table is 7. For pruning settings, the beam size is 200, and 20 translation options are retrieved for each input phrase. The features contained in the baseline translation system are bi-directional phrase translation probabilities, bi-directional lexicalized translation probabilities, bidirectional standard lexicalized reordering probabilities, phrase penalty, word penalty, distance-based reordering model score and language model score. For the hierarchal phrase-based translation model, we use the same setup as the phrase-based model, except we limit the number of symbols on each side of a rule to 5.

[2] http://www.speech.sri.com/projects/srilm/

[3] http://www.statmt.org/moses/giza/GIZA++.html

TABLE II
PERFORMANCE OF THE SVM-BASED CLASSIFIER FOR DIFFERENT COMBINED FEATURES

Combination of Features	Accuracy(%)
PLTP+LenRatio+PV	94.90
SS+TS	84.92
PLTP+LRP+LenRatio+PV	**95.81**
PLTP+LRP+LenRatio+TS+PV	94.98
PLTP+LRP+LenRatio+SS+TS+PV	93.26
PLTP+LRP+Lt/Ls+Ls/Lt+PV	95.06
PLTP+LRP+Lt/Ls+Ls/Lt+TS+PV	94.38

An SVM-based classifier is employed and trained by the widely used open toolkit LIBSVM [31]. There are many choices of the kernel function for the SVM classifier, such as a Linear Function, Sigmoid Function, Polynomial Function and Radial Basis Function (RBF). RBF was chosen as our kernel function after a series of experimental comparisons. This method transforms the space into a high dimension for the non-linear cases and has the advantage of having fewer parameters than the Polynomial Function because the number of parameters determines the complexity of the model directly.

For the training data for the SVM-based classifier, we collect the decoding statistics C and P (Section IV-A) on the optimized weights tuned on the development set. Then, a stratified random sampling set is built, which is meant to randomly sample the same size for the positive and negative data. In total, we use 10,000 positive and 10,000 negative training data instances.

To obtain dependency syntactic features, all of the sentences in the training, development and testing set are parsed using the Berkeley parser[4], which is trained on the Penn Chinese Treebank 6.0, and we then convert the generated phrase-based syntactic trees into dependency trees using the conversion algorithm described in [28], [29].

B. Results on the Phrase-Based Model

To verify the effectiveness of the features used to prune the phrase table under our uniform framework, we compare different feature groups.

Comparison of Various Features: We tuned the classifiers on dev-set by 5-fold cross-validation. It means we divided all the data into 5 parts, 1 of which was used for testing and the rest for training. Each time we chose a different part for testing, and then repeated 5 times. Table II shows the average accuracy. For each feature group, we attempted different kernel functions and parameters on the training data. In this table, PLTP includes phrase and lexicalized phrase translation probabilities, both of which are bi-directional. LRP represents the standard lexicalized reordering probabilities. The syntactic features consist of source syntactic features (SS) and target syntactic features (TS). LenRatio is the length ratio feature. PV represents the statistics significance feature p-value. L_t is the length of the target phrase, and L_s is the length of the source phrase. We also compare the two ratios L_t/L_s and L_s/L_t because the average sentence lengths of Chinese and English are always different empirically.

TABLE III
SIZE AND BLEU FOR DIFFERENT COMBINED FEATURES COMPARED WITH THE BASELINE (POS : NEG = 1 : 1)

	Table Size	NIST'05 BLEU	NIST'06 BLEU
Baseline	15,428,040	25.30	26.86
PLTP+LenRatio+PV	9,355,935(61%)	25.12	**26.98**
SS+TS	8,514,694(55%)	24.65	25.04
PLTP+LRP+LenRatio+PV	6,336,658(41%)	25.33	26.87
PLTP+LRP+LenRatio+TS+PV	**3,756,190(24%)**	25.26	26.73
PLTP+LRP+LenRatio+SS+TS+PV	6,584,166(43%)	25.25	26.87
PLTP+LRP+Ls/Lt+Lt/Ls+PV	5,569,328(36%)	25.34	26.76
PLTP+LRP+Ls/Lt+Lt/Ls+PV+TS	4,962,691(32%)	**25.40**	26.63

The first two groups of features test how the translation probabilities and the syntactic information separately affect the accuracy of the classification. We add the distortion model feature to the third group to see whether the distortion feature is useful. Then, we add the target syntactic information feature after using the source syntactic information.

Table II shows that the combination of PLTP, LRP, LenRatio and PV is very effective and that using the syntactic feature alone is not helpful.

Compared with the predicting capability of the classifiers, we are concerned more with the reduction in the whole translation model and the translation quality. Does a higher accuracy always result in a better performance for the size reduction and BLEU? Table III shows the changes in the translation model sizes and their corresponding BLEU scores.

The results in Table III indicate that the translation quality maintains a relatively consistent trend with the accuracy of the classifier. Additionally, it is better to add the syntactic features because the classifier with syntactic features can discard over 75% of the phrase-pairs.

The results also prove that the combination of PLTP, LRP, LenRatio and PV is very effective in pruning the phrase-pairs, especially the phrases and lexicalized translation probabilities. This result is reasonable because the translation candidates with low translation probabilities are unlikely to be added into the translation lattice. Thus, these pairs can be eliminated without a loss of quality. We have noted that the syntactic features without the addition of the others are weakly supported by the data. Nonetheless, the classifier with the SS + TS features can still remove nearly half of all phrase-pairs. It is not surprising that the BLEU score declines with only the syntactic features. The phrase-based translation model allows illegal syntactic phrases. For this reason, the loss of large illegal syntactic phrases could lead to a reduction in the BLEU score.

In Table III, compared with PLTP + LRP + LenRatio + SS + TS + PV, the features PLTP + LRP + LenRatio + TS + PV are better for pruning, which appears to imply that the source syntactic information has a negative effect on the pruning. We provide an insightful reason for this finding in Section V.

We also compare the different pruned table sizes on the NIST05 and NIST06 data with different combinations of features; these results are shown in Fig. 4. We clearly observe that it is easier to filter table NIST05 (91.7% at most) and NIST06 (91.1% at most) than to prune the whole table (76% at most).

[4] http://code.google.com/p/berkeleyparser

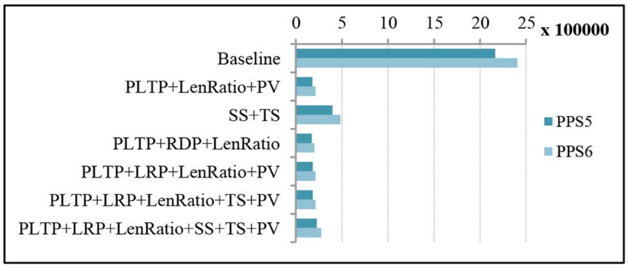


Fig. 4. Changes in the table size of NIST05 and NIST06. We can see that it is better to combine the syntactic features with other features than to use the syntactic features alone.

TABLE IV
BASIC STATISTICS OF THE ORIGINAL AND PRUNED PHRASE TABLE UNDER THE FEATURES PLTP + RDP + LenRatio + TS + PV

	Whole Table		NIST05		NIST06	
	original	pruned	original	pruned	original	pruned
DSN	5,691,593	2,924,778	21,319	19,337	27,911	25,013
ACPS	2.71	1.28	101.5	9.429	86.2	8.52
ALS	4.17	4.04	2.157	2.108	2.23	2.18
ALT	4.02	3.68	3.41	2.12	3.41	2.14

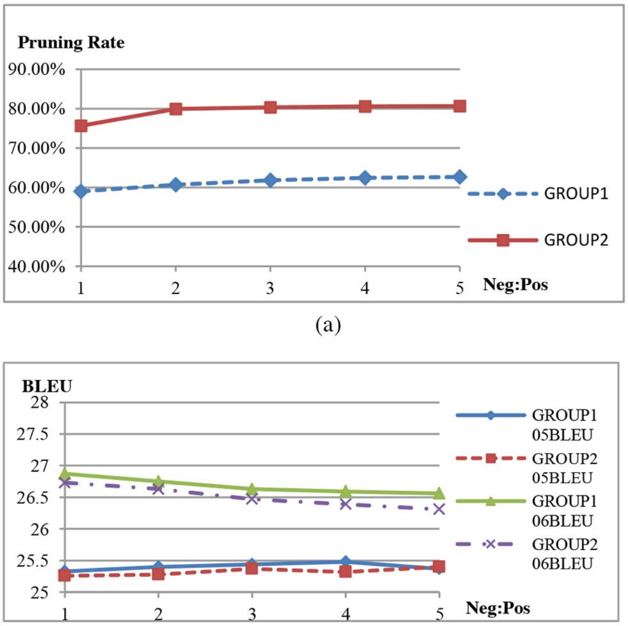
DSN: distinct source phrase numbers ACPS: average candidate-options per distinct source phrase ALS: average length of distinct source phrases ALT: average length of target phrases

To find out why the pruning performance on the testing set is better than on the whole table, we compute statistics about the original tables and the pruned tables with the most effective feature group (PLTP + LRP + LenRatio + TS + PV) in Table IV.

Table IV explicitly reveals the traits of the original table and the pruned table. For the original table of NIST05/06, the average translation options for a distinct source phrase are, in fact, overloaded. Our classifier can cut off most of the options (approximately 80% ~ 90%) and maintain a diverse set of distinct source phrases. While there are few average options for a distinct source phrase in the whole table, we tend to keep the candidate options (only cut off half of the options on average) and prune many useless distinct phrases (approximately 50%). We conclude that the 1-to-many phrase-pairs and the 1-to-few phrase-pairs are distributed differently in the testing set table and the whole table, which is why the pruning rate cannot reach a similar achievement on the testing set table and the whole table.

Another issue from Table IV concerns the lengths of the ratios of the source and target phrases. Empirically, English sentences are longer than Chinese sentences; thus, it looks as though a Chinese phrase should correspond to a longer English phrase. However, the lengths of the well-translated source and target phrases prove to be almost equal, as observed in ALS and ALT in the pruned table. In fact, we obtain a similar result in the positive training data, which is well-translated pairs, where ALS is 1.84 and ALT is 1.98.

Testing on a Different Scale of Negative Data: As mentioned above, we focus on how to prune the phrase table greatly while maintaining the translation quality. Generally, the ratio of the scale of negative and positive data will have a large impact on the classification performance. As we know, having more negative training data means more discarded phrases, but over-pruning could harm the translation quality. Thus, we must find an appropriate balance to trade off the phrase size and the translation quality.



(a)

(b)

Fig. 5. (a) The reduction in the whole table size. When the negative data increases, the size of the table becomes smaller. CR is the abbreviation for the compression rate (b) The translation quality of the testing set with different ratios of negative and positive data.

TABLE V
BLEU SCORES ON THE TEST SET WITH AN INCREASING RATE OF NEGATIVE DATA

POS:NEG	NIST'05/'06	
	GROUP1	GROUP2
Baseline	25.30/26.86	
1:1.00	25.33/ 26.87	25.26 /26.73
1:2.00	25.40 / 26.75	25.28 / 26.63
1:3.00	25.44/ 26.63	25.37/ 26.47
1:4.00	25.48/ 26.59	25.32/ 26.39
1:5.00	25.37/ 26.56	25.40/ 26.31

Table V and Fig. 5 show the different sizes of the translation table and BLEU scores under different ratios (1, 2, 3, 4, and 5). Like experiment setup before, we also randomly sample instances from all training data to build the different ratios of Positive/Negative data. In total, we obtain 10,000 positive and 10,000/20,000/30,000/40,000/50,000 negative training data instances. We choose two groups of features, one of which includes PLTP, LRP, LenRatio and PV because they outperform other combinations in terms of the accuracy in Table II and the translation quality in Table III; the other group of features is obtained by adding TS to Group1 to obtain a better pruning rate that observed in Table III.

We train our classifier to be more sensitive for negative data, which successfully reduces more phrases and retains the BLEU score. With the features in Group2, the whole table can be pruned to a reasonably small size without harming the performance too much. We have found that the syntactic features play a very important role in pruning the phrases. After the ratio of negative and positive data exceeds 2, the highest pruning rate is more than 80% but grows much more slowly.

Thus, we infer that the most suitable ratio is approximately 2, and we choose the negative samples to be double the positive in the following experiment.

When facing a new translation model and new development data, we should determine the best ratio of negative to positive data, which is not randomly chosen. Fig. 5 shows that the translation quality changes very slow; we only need to refer to the trend in the pruning rate curve under a predefined feature combination (see GROUP2). This process is off-line and fast.

Comparison of Various Methods: Table V shows the comparison results of previous methods, especially the hard-rule methods.

TMS is short for the translation model score, which is computed by summing up the weighted bi-directional phrase translation probabilities and the bi-directional lexicalized translation probabilities. By computing all of the TMS scores for all of the phrase-pairs, we obtain a phrase table that ranks the scores from high to low. Then, the phrase-pairs that have a low TMS are pruned when they are lower than the size threshold, e.g., retaining all, retaining 80%, and so on. In our experiments, we used 80%, 60% and 40%. We also compare Histogram pruning, which preserve K options for each distinct source phrase with the highest $p(e|f)$. LenRatio is a way to discard a pair whose length ratio of left-hand to right-hand is outside a given limitation. We also re-implement the method of *usage statistics filtering* (USF), which inspires us to obtain the training data in the way mentioned in Section IV. Then, the popular pruning method of Fisher's test is also performed as a benchmark. Fisher's exact tests in our experiments are performed with SALM [30]. We also compare Relative Entropy, and the combination with Fisher's Test via different interpolation weight α,which is mentioned in [34]. In our experiment, we used a ratio of 1/2 positive to negative data to train the classifier.

Table VI clearly shows that when compared with previous work, our method can provide better translation BLEU scores with a much larger reduction in the translation table. Approximately 80% of the phrase-pairs are discarded by our method with the Group2 features. In this experiment, we use $thr = \alpha - \varepsilon$ as the upper bound of Fisher's test. The study in [32] observed a large number of triple-1 phrase or rules and noted most of them do not occur by chance. A pair whose $C(s)$, $C(t)$ and $C(s,t)$ are all equal to 1 is called a triple-1 pair. In our training corpus, we find that approximately 1/3 of the whole phrase-table is triple-1 phrase-pairs, which is a very large number, but that many of them are useful phrase-pairs.

C. Results on the Hierarchical Phrase-Based Models

Next, we would like to know how effective the method is when it is extended to the hierarchical phrase-based model. Considering the balance of training time and the pruning rate, we set the ratio of negative data to positive data as 2/1.

Pruning Rates and BLEU Scores with Various Features: Table VII shows the model size reduction and translation quality when it is applied to the hierarchical translation model. We have implemented the *Relax-Well-Formed* (RWF) constraints following [19]. RLTP represents the rule and lexicalized translation probabilities. PV is the significance value feature.

TABLE VI
COMPARISON OF PREVIOUS WORK AND OURS

		Table Size	NIST05	NIST06
Baseline		15,428,040	**25.30**	**26.86**
TMS		12,342,438 (80%)	25.16	26.69
		9,256,828 (60%)	24.71	26.36
		6,171,217 (40%)	24.14	26.24
Histogram	Options=10	11,403,487(74%)	25.18	26.79
	Options=5	10,185,880(66%)	24.74	26.26
	Options=2	7,853,380(51%)	23.56	23.68
LenRatio	1/3	13,542,808 (88%)	25.07	25.75
	2/3	9,173,848 (60%)	17.86	17.34
USF		13,034,347 (84%)	24.92	26.67
Fisher's Test($thr = \alpha - \varepsilon$)		6,085,910 (39%)	25.05	26.14
Relative Entropy	τ =0.002065	12,342,432(80%)	25.29	26.83
	τ =0.031112	9,242,094(60%)	25.31	26.75
	τ =0.087995	6,171,214(40%)	25.43	26.78
	τ =0.219705	3,041,640(20%)	24.57	25.81
Fisher's Test with Relative Entropy	α =0.8	9,256,823（60%）,*thr*= 0.0249	25.30	26.68
		6,172,516（40%）,*thr*= 0.0687	25.33	26.55
		3,085,600（20%）,*thr*=0.1667	24.68	26.04
	α =0.6	9,218,792（60%）,*thr*= 0.0187	25.30	26.70
		6,171,214（40%）,*thr*= 0.0523	25.39	26.74
		3,085,608（20%）,*thr*=0.1299	24.66	25.93
	α =0.4	9,225,020（60%）,*thr*= 0.0124	25.30	26.71
		6,171,211（40%）,*thr*= 0.0349	25.40	26.77
		3,012,358（20%）,*thr*=0.0879	24.64	25.87
	α =0.2	9,240,457（60%）,*thr*= 0.0062	25.31	26.71
		6,171,190（40%）,*thr*= 0.0175	25.39	26.78
		3,020,772（20%）,*thr*=0.0439	24.65	25.87
Ours (POS/NEG=1/2)	GROUP1	**6,062,743 (39%)**	**25.40**	**26.75**
	GROUP2	**3,094,854 (20%)**	**25.28**	**26.63**

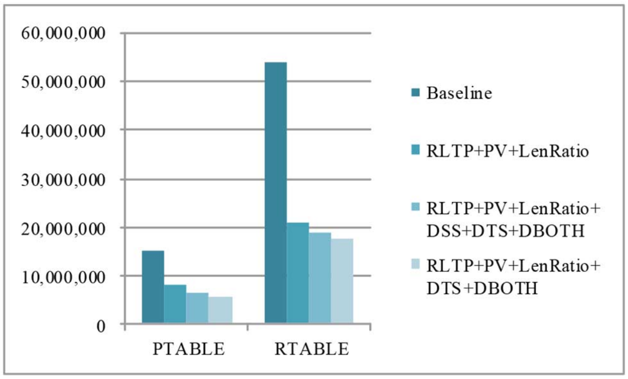


Fig. 6. Details of PTABLE (Phrase-table) and RTABLE (Rule-table).

LenRatio is the length ratio feature. We use DSS, DTS, and DBOTH to represent the dependency syntactic feature of the source, target, and both sides, respectively. Fig. 6 gives the pruning details of the phrase-table and rule-table in the hierarchical translation model.

Table VII tells us that the reduction in the whole size is approximately 67%, which shows that size of the rule-table is sharply reduced while the translation performance still maintains at a relatively stable level. The comparison of features suggests that the dependency syntactic features are helpful and that the system works better when we exclude the source side

TABLE VII
SIZE AND BLEU OF DIFFERENT COMBINED FEATURES IN THE HIERARCHICAL MODEL

	Table Size	BLEU	
		NIST05	NIST06
Baseline	69,393,064(100%)	25.58	28.15
RWF	45,338,911(65%)	**25.91**	28.11
RLTP+PV+LenRatio	28,614,096 (41%)	25.73	28.00
RLTP+PV+LenRatio +DSS+DTS+DBOTH	24,954,654 (36%)	25.77	28.05
RLTP+PV+LenRatio +DTS+DBOTH	**23,126,972 (33%)**	25.69	**28.15**

TABLE VIII
ACCURACY OF CLASSIFIERS WITH DIFFERENT FEATURE COMBINATIONS

	Accuracy	
	P-Classifier	R-Classifier
RLTP+PV+LenRatio	98.07%	95.99%
RLTP+PV+LenRatio+DSS+DTS+DBOTH	96.97%	95.24%
RLTP+PV+LenRatio+DTS+DBOTH	97.21%	95.42%

P-classifier is for the phrase-table; R-classifier is for the rule-table.

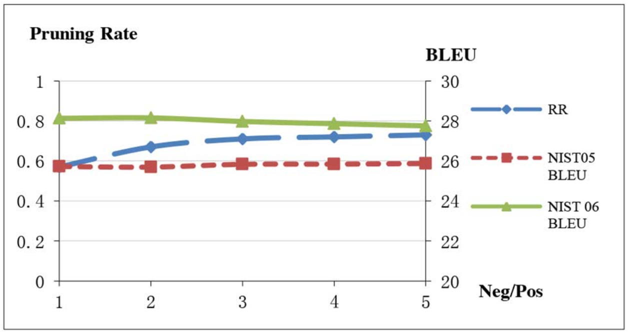


Fig. 7. The reduction in the whole table size with different scales of negative data in the hierarchical model.

of the dependency syntactic features. An interesting observation is that we obtain a similar result in the previous experiment for phrase-based models (Table III). We provide a more detailed analysis on such results in the next section.

Accuracy Details of Different Classifiers and Feature Settings: Tables VII and VIII show a relatively consistent trend between the accuracy and translation quality, which is similar to the experimental results on the phrase-based models.

Testing on a Different Scale of Negative Data: The pruning rates and BLEU scores for different scales of negative data are given in Fig. 7. When the amount of negative data increases, the model pruning rate (RR) becomes higher. We can find that the best ratio point for this model is approximately 3, where approximately 70% of the rules can be filtered, and the BLEU Scores of NIST'05/'06 are 25.83/27.97, which changes slightly.

D. Results on Additional Experiments

All of the features that we used are directly combined into the classification framework for pruning; in fact, we doubt whether they can improve the translation quality by adding them (including PV and LenRatio) into a baseline translation system as additional log-linear features. Thus, we perform additional experiments to test this idea, and we test whether the TMS can be improved in this setup. In order to make the results stable, we vary initial model parameters and report the average BLEU scores. The experimental results are shown in Table IX. The new baseline BLEU scores of both sets improve compared with the original baseline scores (25.30 for NIST05 and 26.86 for NIST06). And performance of our method still outperforms TMS which loses about 2 points at the 60% pruning rate, while ours change subtle at 80% pruning rate.

TABLE IX
COMPARISON OF THE EXPERIMENTAL RESULTS USING THE FEATURES IN A CLASSIFICATION FRAMEWORK OR DIRECTLY ADDING TO THE DECODER

	Whole Table	NIST05	NIST06
Baseline#	15,428,040	25.33	27.31
TMS#	12,342,434 (80%)	25.08	27.00
	9,256,825 (60%)	24.64	26.51
	6,171,218 (40%)	23.53	25.72

We use Baseline# to distinguish from Baseline before the results. Additional features are involved in this translation system. TMS# calculates a composite score of the additional features aside from the features of TMS in Table VI.

V. FURTHER ANALYSIS AND DISCUSSION

A. Are Target Syntactic Features Better Than Source Features?

As observed from the experimental results above, when we remove source syntactic features from the full syntactic information, the pruning rate becomes higher under both the phrase-based and hierarchical phrase-based situations. Thus, we have to guess that the syntactic feature on the target side is more effective than the syntactic feature on the source side. In the following, we provide a detailed analysis.

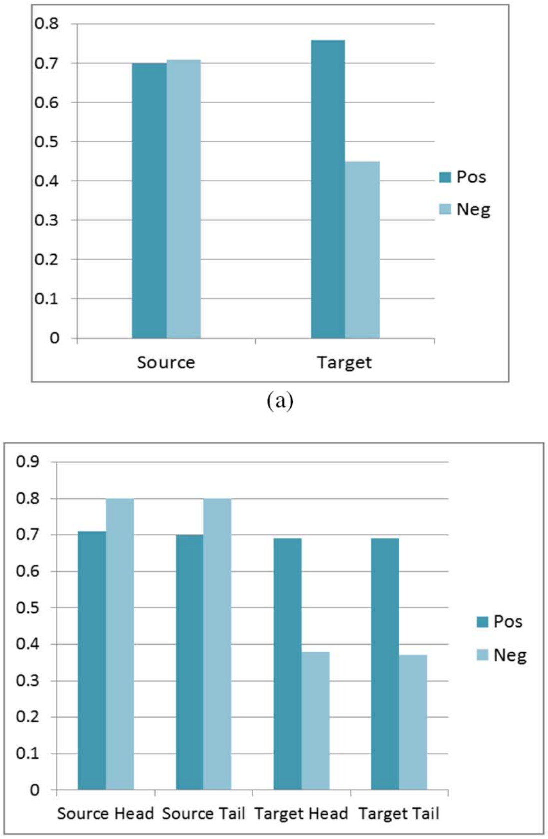
We introduce the *Average Syntactic Well-Formed* (ASW) samples to test our hypothesis.

Let V represents the value of the syntactic feature of a rule or a phrase. Here, V is a binary value in the phrase-based model but a real value in the hierarchical phrase-based models. Then, $\text{ASW} = E(V)$. It is obvious that a larger ASW divergence between the positive and negative data results in better performance.

Fig. 8 shows the ASW between the positive data and the negative data. This figure clearly indicates that the *ASW* divergence on the target side is more evident than that of the source side.

Our original aim of adding the source syntactic features was to select the good phrase-pairs or rules that have legal syntactic or well-formed structures. Unexpectedly, from Fig. 8(b), we see that the negative data of the source side in the phrase-based model contains more legal syntactics, which leads to a bad effect on the pruning rate, as shown in Table III (from 24% to 43%). A similar result is also shown in Table VI (from 33% to 36%).

From the analysis above, we can conclude that for the Chinese-to-English translation task, it is better not to use the source syntactic information as a single feature. However, whether the source syntactic information has the same effect in the opposite translation direction is to be tested in the future. Considering the space constraints, we do not perform this experiment in this paper. If future work gives a contrary result, i.e., the target syntactic information is useless in the English-to-Chinese translation task, then we can see that the syntactic information is more closely related to the language itself than to the translation direction. Additionally, the error rates of the parsers will be taken into account.



(a)

(b)

Fig. 8. (a) *ASW* of the source and target side in the rule-table in hierarchical training samples (b) *ASW* of the source and target side in the phrase-table in phrase-based training samples. Source Head refers to the count of the legal syntactic source head (SHS), and Source Tail refers to that of STS, which is similar to the Target Head and Target Tail.

TABLE X
PHRASE-TABLE DETAILS OF EACH METHOD BEFORE AND AFTER THE PRUNING PROCESS

	DSN	CR	CRM	CRR	ACPS	ALS	ALT
Before Pruning	5,691,593	-------	-------	-------	2.71	4.17	4.02
Fisher's Test	3,665,568	61%	86%	39%	1.66	4.38	4.06
Relative Entropy	3,936,671	60%	89%	35%	1.56	3.97	3.87
Fisher's Test with Relative Entropy	3,934,724	60%	90%	35%	1.57	3.98	3.89
TMS	4,033,569	60%	78%	45%	1.53	4.15	4.14
Histogram	5,691,593	49%	90%	15%	1.79	4.17	4.21
USF	5,691,593	16%	36%	1%	2.29	4.17	4.11
Ours (G2,1:1)	2,924,778	74%	**92%**	**62%**	1.28	4.04	3.68

DSN: distinct source phrase numbers CR: compression rate CRM: compression rate of 1-to-many ($>$ 5) phrases CRR: compression rate of 1-to-few ($<=$ 5) phrases ACPS: average candidate-options per distinct source phrase ALS: average length of distinct source phrases ALT: average length of target phrases

B. Discussion About What Types of Phrases will be Discarded

Each pruning method shows a different bias on discarded phrase-pairs due to different theoretical characteristics of each method. Table X shows the details of the phrase-table for each method. In the table, the threshold of Fisher's significance test is $\alpha - \varepsilon$, the threshold of TMS is 40%, the features of our methods are GROUP2 features (PLTP+LRP+LenRatio+TS+PV), and the scales of negative and positive training data are the same.

For Fisher's significance test, if the source and target phrases have no significant association, then they are discarded. Thus, many accidentally extracted translation options are cut off, which causes a large pruning rate of 1-to-many phrase-pairs.

TMS pruning will discard many translation options with low-weighted translation probabilities, both for 1-to-many and 1-to-few phrase-pairs.

The histogram pruning method leaves out many translation options that have low phrase translation probabilities; thus, the 1-to-many phrase-pairs are pruned heavily. However, the pruned table is still large because no distinct source phrase is discarded.

The USF method is good at selecting very effective phrase-pairs in theory, but only when the development set is sufficient can the whole table become pruned substantially.

Our method takes advantage of composite factors; it achieves the best performance on both 1-to-many and 1-to-few translation pairs.

VI. RELATED WORK

For the phrase-based translation model, one of the typical pruning methods is based on the usage of statistics during the decoding of development data [5], which could prune most of the phrase-pairs without a significant loss in the quality of the PBTS. A similar method is to extract phrase-pairs using a scoring metric [6]. However, these methods absolutely depend on the statistics of the development decoding, without any linguistic information. This arrangement means that a large amount of development data is required to address an enormous translation model, and the result involves some inevitable loss of good linguistic translation pairs.

Another effective method integrated in Moses uses the significance test of phrase-pair co-occurrences, which prunes phrase-pairs using a "p-value" [7]. Similar work based on statistical independence uses "Noise" as the filtering criterion [8]. Besides, some approaches of computing relative entropy [33], [34] are good at pruning redundant phrases, especially for those long phrases that can be replaced by shorter phrases. However, all these methods mentioned above ignore the syntactic information of phrases and sometimes discard the potential useful phrases that occur with a low probability in the training corpus. The paper in [9] uses strict syntactic constraints to prune most of the phrases. For that application, most of the phrase-pairs violate the strict constraints, and many of them are well-translated pairs that can contribute to the final translation; thus, the strict syntactic constraints lead to a large reduction in the translation quality. Nonetheless, it has been proven that using relaxed syntactic constraints in the source phrase can filter some phrases without harming the quality [10]. However, little is mentioned about why the target syntactic information should not be used, and the pruning capacity still has room to be improved. Some other work uses syntax to improve word alignments, which results in a better phrase-table [11]–[14]. These authors can improve the translation quality, but the model size still remains large because they concentrate more on decreasing the error propagation instead of on pruning. The study in [15] introduces a method to filter the term-pairs validation as a classification problem that mainly focuses on the newly extracted translation lexicons for word alignment, but it employs only manual tagging data and mainly aims at those term-pairs that have good linguistic value for alignment, regardless of the translation table size and translation quality. Another similar study, in [16], employs one-class SVM to help select good phrase-pairs. Our idea is inspired by [16]. The major differences lie in feature selection and training data building: 1) Our underlying principle is to compress a table maximally while retaining comparable translation quality. The study in [16] concentrates only on improving the BLEU score; 2) We highlight and use the syntactic and dependency-syntactic structure information of a phrase-pair or rule; and 3) We propose a simple but effective way to automatically generate the training data independent from the selected features. The method in [16] for obtaining negative training data that is closely related to the features and the whole process of obtaining the training data is relatively complicated. Their work focuses on how to improve the BLEU scores by selecting good translation-pairs. They adopt oracle decoding methods to obtain positive data and then use an iterative mapping convergence process to obtain negative data. However, they do not care about the size of the pruned table.

For the hierarchical phrase-based translation model, [17] extends the significance test in [7] to the hierarchical phrase-based model. Because phrases are more sparse than rules with non-terminals in general, the power of filtering in [7] is somewhat weak. Apart from these studies, other well-known methods are based on syntactic information. Most rules in the hierarchical phrase-based translation model can be discarded under the constraint of having a well-formed structure on the target side [18]. However, the experimental results in [19] indicate that the methods in [18] lead to a significant degradation in the translation quality. Instead, [19] proposes a *Relax-Well-Formed* method to improve the translation quality, with fewer rules being discarded.

VII. Conclusions

In this paper, we propose a unified framework to perform a pruning task on both phrase-based and hierarchical phrase-based translation models, which greatly reduces the model size while retaining the translation quality.

In summary, our methods have the following advantages over existing studies. (1) We exploit and integrate all of the heuristic features in a classification framework, which makes it more robust and effective to filter the phrase-pairs or rules. The experimental results have shown that it is effective at pruning approximately 80% of the phrase-pairs and 70% of the rules without harming the translation quality. (2) The training process of our classifier is embedded off-line in our framework and is independent of the test set (Fisher is also independent). In our classifier, only two factors are determined. One factor is the feature selection, and the other factor is the ratio of positive to negative data. Both of these factors can attain the best combination under our framework. (3) Training data of the classifier are generated automatically from the decoding path with a tuned translation model and development data. (4) Our unified framework breaks the limitation of specific translation model pruning, which makes it possible to extend and transfer to other syntactic translation models.

In future work, we will study the method using other language pairs and use other linguistic features according to the characteristics of different languages. Moreover, we wish to use our pruning framework embedded in other translation models, such as syntax-based models.

References

[1] D. Chiang, "Hierarchical phrase-based translation," *Comput. Linguist.*, vol. 33, pp. 201–228, 2007.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, and R. Zens, "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL Demo and Poster Sessions*, Prague, Czech Republic, Jun. 2007, pp. 177–180.

[3] F. J. Och and H. Ney, "The alignment template approach to statistical machine translation," *Comput. Linguist.*, vol. 30, pp. 417–449, 2004.

[4] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Comput. Linguist.*, vol. 29, pp. 19–51, 2003.

[5] M. Eck, S. Vogel, and A. Waibel, "Translation model pruning via usage statistics for statistical machine translation," in *Proc. HLT-NAACL (Short Papers)*, Rochester, NY, USA, Apr. 2007, pp. 21–24.

[6] L. S. Zettlemoyer and R. C. Moore, "Selective phrase pair extraction for improved statistical machine translation," in *Proc. Conf. North Amer. Chap. Assoc. Comput. Linguist.*, 2007, pp. 209–212.

[7] J. H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving translation quality by discarding most of the phrasetable," in *Proc. (EMNLP-CoNLL)*, Prague, Czech Republic, 2007, pp. 967–975.

[8] N. Tomeh, N. Cancedda, and M. Dymetman, "Complexity-based phrase-table filtering for statistical machine translation," in *Proc. MT Summit XII*, Ottawa, ON, Canada, Aug. 2009.

[9] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. Conf. North Amer. Chap. Assoc. Comput. Linguist. Human Lang. Technol.*, Stroudsburg, PA, USA, vol. 1, pp. 48–54.

[10] H. Cao, A. Finch, and E. Sumita, "Syntactic Constraints on Phrase Extraction for Phrase-Based Machine Translation," in *Proc. SSST-4, 4th Workshop Syntax and Structure in Statist. Translat.*, Beijing, China, Aug. 2010, pp. 28–33.

[11] Y. Ma, S. Ozdowska, Y. Sun, and A. Way, "Improving word alignment using syntactic dependencies," in *Proc. 2nd ACL Workshop Syntax and Structure in Statist. Translat.*, Columbus, OH, USA, Jun. 2008, pp. 69–77.

[12] C. Cherry and D. Lin, "Soft syntactic constraints for word alignment through discriminative training," in *Proc. COLING/ACL Main Conf. Poster Sessions*, Sydney, Australia, Jul. 2006, pp. 105–112.

[13] J. DeNero and D. Klein, "Tailoring word alignments to syntactic machine translation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguist.*, Prague, Czech Republic, Jun. 2007, pp. 17–24.

[14] U. Hermjakob, "Improved word alignment with statistics and linguistic heuristics," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Singapore, Aug. 6–7, 2009, pp. 229–237.

[15] K. Kavitha, L. Gomes, and G. P. Lopes, "Using SVMs for Filtering Translation Tables for Parallel Corpora Alignment," in *Proc. EPIA*, 2011.

[16] N. Tomeh, M. Turchi, G. Wisinewski, A. Allauzen, and F. Yvon, "How Good Are Your Phrases? Assessing Phrase Quality with Single Class Classification," in *Proc. Int. Workshop Spoken Lang. Translat.*, San Francisco, CA, USA, Dec. 2011, pp. 261–268.

[17] M. Yang and J. Zheng, "Toward smaller, faster, and better hierarchical phrase-based SMT," in *Proc. ACL-IJCNLP Conf. Short Papers*, Stroudsburg, PA, USA, 2009, pp. 237–240.

[18] L. Shen, J. Xu, and R. Weischedel, "A new string-to-dependency machine translation algorithm with a target dependency language model," in *Proc. ACL-08: HLT*, 2008, pp. 577–585.

[19] Z. Wang, Y. Lü, Q. Liu, and Y. S. Hwang, "Better filtration and augmentation for hierarchical phrase-based translation rules," in *Proc. ACL Conf. Short Papers*, Uppsala, Sweden, Jul. 11–16, 2010, pp. 142–146.

[20] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist.*, Philadelphia, PA, USA, Jul. 2002, pp. 311–318.

[21] L. Cui, D. Zhang, M. Li, M. Zhou, and T. Zhao, "A joint rule selection model for hierarchical phrase-based translation," in *Proc. ACL Conf. Short Papers*, Uppsala, Sweden, Jul. 2010, pp. 6–11.

[22] Z. Huang, M. Čmejrek, and B. Zhou, "Soft syntactic constraints for hierarchical phrase-based translation using latent syntactic distributions," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Oct. 9–11, 2010, pp. 138–147.

[23] G. Iglesias, A. de Gispert, E. R. Banga, and W. Byrne, "Rule filtering by pattern for efficient hierarchical translation," in *Proc. 12th Conf. Eur. Chap. ACL*, Athens, Greece, Mar. Apr. 30 3, 2009, pp. 380–388.

[24] D. Chiang, K. Knight, and W. Wang, "11,001 new features for statistical machine translation," in *Proc. Human Lang. Technol. : Annu. Conf. North Amer. Chap. ACL*, Boulder, CO, USA, Jun. 2009, pp. 218–226.

[25] Y. Marton and P. Resnik, "Soft syntactic constraints for hierarchical phrased-based translation," in *Proc. ACL-08: HLT*, Columbus, OH, USA, Jun. 2008, pp. 1003–1011.

[26] L. Shen, J. Xu, B. Zhang, S. Matsoukas, and R. Weischedel, "Effective use of linguistic and contextual information for statistical machine translation," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Singapore, Aug. 6–7, 2009, pp. 72–80.

[27] K. Imamura, "Application of translation knowledge acquired by hierarchical phrase alignment for pattern-based MT," in *Proc. TMI*, 2002, pp. 74–84.

[28] Z. W. a. C. Zong, "Reranking based on higher-order lexical dependencies," in *Proc. 5th Int. Joint Conf. Nat. Lang. Process.*, Chiang Mai, Thailand, Nov. 8–13, 2011, pp. 1251–1259.

[29] Z. Wang and C. Zong, "Phrase structure parsing with dependency structure," in *Proc. Coling: Poster Vol.*, Beijing, China, Aug. 2010, pp. 1292–1300.

[30] Y. Zhang and S. Vogel, "Suffix array and its applications in empirical natural language processing," Lang. Technol. Inst., School of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-LTI-06-010, Dec. 2006.

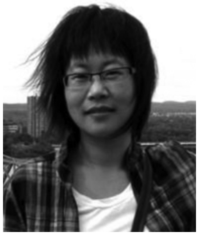
[31] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 2, p. 27, 2011.

[32] R. C. Moore, "On log-likelihood-ratios and the significance of rare events," in *Proc. EMNLP*, Barcelona, Spain, 2004, pp. 333–340.

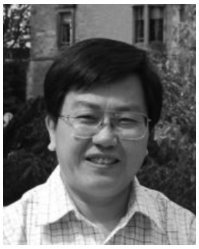
[33] R. Zens, D. Stanton, and P. Xu., "A systematic comparison of phrase table pruning techniques," in *Proc. Joint Conf. Empir. Meth. Nat. Lang. Process. Comput. Nat. Lang. Learn.*, Jeju Island, Korea, Jul. 2012, pp. 972–983.

[34] L. Wang, T. Nadi, X. Guang, B. Alan, and T. Isabel, "Improving relative-entropy pruning using statistical significance," in *Proc. 25th Int. Conf. Comput. Linguist. (Posters)*, Mumbai, India, Dec. 2012, pp. 713–722.

Mei Tu is a Ph.D. She graduated from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in 2015. Her research interests include machine translation and the key techniques of natural language processing.



Yu Zhou is an Associate Professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Her research interests include machine translation and the key techniques of natural language processing.



Chengqing Zong received his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in March 1998. He is a Professor at the National Laboratory of Pattern Recognition, Chinese Academy of Sciences' Institute of Automation. His research interests include machine translation, natural language processing, and sentiment classification. He is a member of International Committee on Computational Linguistics (ICCL). He is associate editor of *ACM Transactions on Asian and Low-Resource Language Information Processing* (*TALLIP*) and editorial board member of IEEE *Intelligent Systems*, *Machine Translation*, *Journal of Computer Science and Technology*. Also, he served ACL-IJCNLP 2015 as a PC co-chair.