Cross Language Dependency Parsing using a Bilingual Lexicon*

Hai Zhao(赵海)†‡, Yan Song(宋彦)†, Chunyu Kit†, Guodong Zhou‡
†Department of Chinese, Translation and Linguistics
City University of Hong Kong
83 Tat Chee Avenue, Kowloon, Hong Kong, China
‡School of Computer Science and Technology
Soochow University, Suzhou, China 215006
{haizhao,yansong,ctckit}@cityu.edu.hk, gdzhou@suda.edu.cn

Abstract

This paper proposes an approach to enhance dependency parsing in a language by using a translated treebank from another language. A simple statistical machine translation method, word-by-word decoding, where not a parallel corpus but a bilingual lexicon is necessary, is adopted for the treebank translation. Using an ensemble method, the key information extracted from word pairs with dependency relations in the translated text is effectively integrated into the parser for the target language. The proposed method is evaluated in English and Chinese treebanks. It is shown that a translated English treebank helps a Chinese parser obtain a state-of-the-art result.

1 Introduction

Although supervised learning methods bring state-of-the-art outcome for dependency parser inferring (McDonald et al., 2005; Hall et al., 2007), a large enough data set is often required for specific parsing accuracy according to this type of methods. However, to annotate syntactic structure, either phrase- or dependency-based, is a costly job. Until now, the largest treebanks[1] in various languages for syntax learning are with around one million words (or some other similar units). Limited data stand in the way of further performance enhancement. This is the case for each individual language at least. But, this is not the case as we observe all treebanks in different languages as a whole. For example, of ten treebanks for CoNLL-2007 shared task, none includes more than 500K tokens, while the sum of tokens from all treebanks is about two million (Nivre et al., 2007).

As different human languages or treebanks should share something common, this makes it possible to let dependency parsing in multiple languages be beneficial with each other. In this paper, we study how to improve dependency parsing by using (automatically) translated texts attached with transformed dependency information. As a case study, we consider how to enhance a Chinese dependency parser by using a translated English treebank. What our method relies on is not the close relation of the chosen language pair but the similarity of two treebanks, this is the most different from the previous work.

Two main obstacles are supposed to confront in a cross-language dependency parsing task. The first is the cost of translation. Machine translation has been shown one of the most expensive language processing tasks, as a great deal of time and space is required to perform this task. In addition, a standard statistical machine translation method based on a parallel corpus will not work effectively if it is not able to find a parallel corpus that right covers source and target treebanks. However, dependency parsing focuses on the relations of word pairs, this allows us to use a dictionary-based translation without assuming a parallel corpus available, and the training stage of translation may be ignored and the decoding will be quite fast in this case. The second difficulty is that the outputs of translation are hardly qualified for the parsing purpose. The most challenge in this aspect is morphological preprocessing. We regard that the morphological issue should be handled aiming at the specific language, our solution here is to use character-level features for a target language like Chinese.

The rest of the paper is organized as follows. The next section presents some related existing work. Section 3 describes the procedure on tree-

The study is partially supported by City University of Hong Kong through the Strategic Research Grant 7002037 and 7002388. The first author is sponsored by a research fellowship from CTL, City University of Hong Kong.

[1] It is a tradition to call an annotated syntactic corpus as treebank in parsing community.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 55–63,
Suntec, Singapore, 2-7 August 2009.

bank translation and dependency transformation. Section 4 describes a dependency parser for Chinese as a baseline. Section 5 describes how a parser can be strengthened from the translated treebank. The experimental results are reported in Section 6. Section 7 looks into a few issues concerning the conditions that the proposed approach is suitable for. Section 8 concludes the paper.

2 The Related Work

As this work is about exploiting extra resources to enhance an existing parser, it is related to domain adaption for parsing that has been draw some interests in recent years. Typical domain adaptation tasks often assume annotated data in new domain absent or insufficient and a large scale unlabeled data available. As unlabeled data are concerned, semi-supervised or unsupervised methods will be naturally adopted. In previous works, two basic types of methods can be identified to enhance an existing parser from additional resources. The first is usually focus on exploiting automatic generated labeled data from the unlabeled data (Steedman et al., 2003; McClosky et al., 2006; Reichart and Rappoport, 2007; Sagae and Tsujii, 2007; Chen et al., 2008), the second is on combining supervised and unsupervised methods, and only unlabeled data are considered (Smith and Eisner, 2006; Wang and Schuurmans, 2008; Koo et al., 2008).

Our purpose in this study is to obtain a further performance enhancement by exploiting treebanks in other languages. This is similar to the above first type of methods, some assistant data should be automatically generated for the subsequent processing. The differences are what type of data are concerned with and how they are produced. In our method, a machine translation method is applied to tackle golden-standard treebank, while all the previous works focus on the unlabeled data.

Although cross-language technique has been used in other natural language processing tasks, it is basically new for syntactic parsing as few works were concerned with this issue. The reason is straightforward, syntactic structure is too complicated to be properly translated and the cost of translation cannot be afforded in many cases. However, we empirically find this difficulty may be dramatically alleviated as dependencies rather than phrases are used for syntactic structure representation. Even the translation outputs are not so good as the expected, a dependency parser for the target language can effectively make use of them by only considering the most related information extracted from the translated text.

The basic idea to support this work is to make use of the semantic connection between different languages. In this sense, it is related to the work of (Merlo et al., 2002) and (Burkett and Klein, 2008). The former showed that complementary information about English verbs can be extracted from their translations in a second language (Chinese) and the use of multilingual features improves classification performance of the English verbs. The latter iteratively trained a model to maximize the marginal likelihood of tree pairs, with alignments treated as latent variables, and then jointly parsing bilingual sentences in a translation pair. The proposed parser using features from monolingual and mutual constraints helped its log-linear model to achieve better performance for both monolingual parsers and machine translation system. In this work, cross-language features will be also adopted as the latter work. However, although it is not essentially different, we only focus on dependency parsing itself, while the parsing scheme in (Burkett and Klein, 2008) based on a constituent representation.

Among of existing works that we are aware of, we regard that the most similar one to ours is (Zeman and Resnik, 2008), who adapted a parser to a new language that is much poorer in linguistic resources than the source language. However, there are two main differences between their work and ours. The first is that they considered a pair of sufficiently related languages, Danish and Swedish, and made full use of the similar characteristics of two languages. Here we consider two quite different languages, English and Chinese. As fewer language properties are concerned, our approach holds the more possibility to be extended to other language pairs than theirs. The second is that a parallel corpus is required for their work and a strict statistical machine translation procedure was performed, while our approach holds a merit of simplicity as only a bilingual lexicon is required.

3 Treebank Translation and Dependency Transformation

3.1 Data

As a case study, this work will be conducted between the source language, English, and the target language, Chinese, namely, we will investigate

how a translated English treebank enhances a Chinese dependency parser.

For English data, the Penn Treebank (PTB) 3 is used. The constituency structures is converted to dependency trees by using the same rules as (Yamada and Matsumoto, 2003) and the standard training/development/test split is used. However, only training corpus (sections 2-21) is used for this study. For Chinese data, the Chinese Treebank (CTB) version 4.0 is used in our experiments. The same rules for conversion and the same data split is adopted as (Wang et al., 2007): files 1-270 and 400-931 as training, 271-300 as testing and files 301-325 as development. We use the gold standard segmentation and part-of-speech (POS) tags in both treebanks.

As a bilingual lexicon is required for our task and none of existing lexicons are suitable for translating PTB, two lexicons, LDC Chinese-English Translation Lexicon Version 2.0 (LDC2002L27), and an English to Chinese lexicon in StarDict[2], are conflated, with some necessary manual extensions, to cover 99% words appearing in the PTB (the most part of the untranslated words are named entities.). This lexicon includes 123K entries.

3.2 Translation

A word-by-word statistical machine translation strategy is adopted to translate words attached with the respective dependency information from the source language to the target one. In detail, a word-based decoding is used, which adopts a log-linear framework as in (Och and Ney, 2002) with only two features, translation model and language model,

$$P(c|e) = \frac{\exp[\sum_{i=1}^{2} \lambda_i h_i(c,e)]}{\sum_c \exp[\sum_{i=1}^{2} \lambda_i h_i(c,e)]}$$

Where

$$h_1(c,e) = \log(p_\gamma(c|e))$$

is the translation model, which is converted from the bilingual lexicon, and

$$h_2(c,e) = \log(p_\theta(c))$$

is the language model, a word trigram model trained from the CTB. In our experiment, we set two weights $\lambda_1 = \lambda_2 = 1$.

[2]StarDict is an open source dictionary software, available at http://stardict.sourceforge.net/.

The conversion process of the source treebank is completed by three steps as the following:
1. Bind POS tag and dependency relation of a word with itself;
2. Translate the PTB text into Chinese word by word. Since we use a lexicon rather than a parallel corpus to estimate the translation probabilities, we simply assign uniform probabilities to all translation options. Thus the decoding process is actually only determined by the language model. Similar to the "bag translation" experiment in (Brown et al., 1990), the candidate target sentences made up by a sequence of the optional target words are ranked by the trigram language model. The output sentence will be generated only if it is with maximum probability as follows,

$$\begin{aligned} c &= \text{argmax}\{p_\theta(c)p_\gamma(c|e)\} \\ &= \text{argmax}\ p_\theta(c) \\ &= \text{argmax}\prod p_\theta(w_c) \end{aligned}$$

A beam search algorithm is used for this process to find the best path from all the translation options; As the training stage, especially, the most time-consuming alignment sub-stage, is skipped, the translation only includes a decoding procedure that takes about 4.5 hours for about one million words of the PTB in a 2.8GHz PC.
3. After the target sentence is generated, the attached POS tags and dependency information of each English word will also be transferred to each corresponding Chinese word. As word order is often changed after translation, the pointer of each dependency relationship, represented by a serial number, should be re-calculated.

Although we try to perform an exact word-by-word translation, this aim cannot be fully reached in fact, as the following case is frequently encountered, multiple English words have to be translated into one Chinese word. To solve this problem, we use a policy that lets the output Chinese word only inherits the attached information of the highest syntactic head in the original multiple English words.

4 Dependency Parsing: Baseline

4.1 Learning Model and Features

According to (McDonald and Nivre, 2007), all data-driven models for dependency parsing that have been proposed in recent years can be described as either graph-based or transition-based.

Table 1: Feature Notations

Notation	Meaning
s	The word in the top of stack
s'	The first word below the top of stack.
s_{-1},s_1...	The first word before(after) the word in the top of stack.
i, i_{+1},...	The first (second) word in the unprocessed sequence, etc.
dir	Dependent direction
h	Head
lm	Leftmost child
rm	Rightmost child
rn	Right nearest child
form	word form
pos	POS tag of word
cpos1	coarse POS: the first letter of POS tag of word
cpos2	coarse POS: the first two POS tags of word
lnverb	the left nearest verb
$char_1$	The first character of a word
$char_2$	The first two characters of a word
$char_{-1}$	The last character of a word
$char_{-2}$	The last two characters of a word
.	's, i.e., '*s.dprel*' means dependent label of character in the top of stack
+	Feature combination, i.e., '*s.char+i.char*' means both *s.char* and *i.char* work as a feature function.

Although the former will be also used as comparison, the latter is chosen as the main parsing framework by this study for the sake of efficiency. In detail, a shift-reduce method is adopted as in (Nivre, 2003), where a classifier is used to make a parsing decision step by step. In each step, the classifier checks a word pair, namely, *s*, the top of a stack that consists of the processed words, and, *i*, the first word in the (input) unprocessed sequence, to determine if a dependent relation should be established between them. Besides two dependency arc building actions, a shift action and a reduce action are also defined to maintain the stack and the unprocessed sequence. In this work, we adopt a left-to-right arc-eager parsing model, that means that the parser scans the input sequence from left to right and right dependents are attached to their heads as soon as possible (Hall et al., 2007).

While memory-based and margin-based learning approaches such as support vector machines are popularly applied to shift-reduce parsing, we apply maximum entropy model as the learning model for efficient training and adopting overlapped features as our work in (Zhao and Kit, 2008), especially, those character-level ones for Chinese parsing. Our implementation of maximum entropy adopts L-BFGS algorithm for parameter optimization as usual.

With notations defined in Table 1, a feature set as shown in Table 2 is adopted. Here, we explain some terms in Tables 1 and 2. We used a large scale feature selection approach as in (Zhao et al., 2009) to obtain the feature set in Table 2. Some feature notations in this paper are also borrowed from that work.

The feature *curroot* returns the root of a partial parsing tree that includes a specified node. The feature *charseq* returns a character sequence whose members are collected from all identified children for a specified word.

In Table 2, as for concatenating multiple substrings into a feature string, there are two ways, seq and bag. The former is to concatenate all substrings without do something special. The latter will remove all duplicated substrings, sort the rest and concatenate all at last.

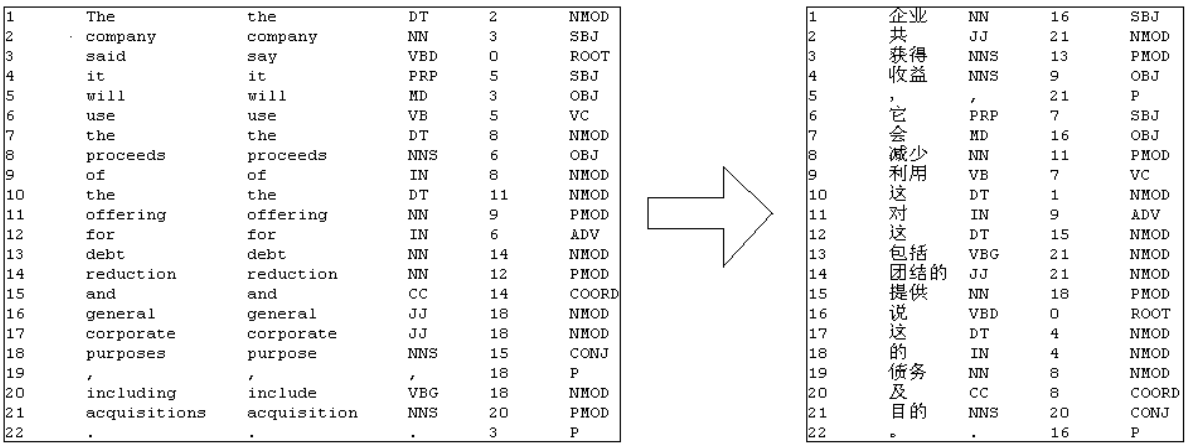
Note that we systemically use a group of character-level features. Surprisingly, as to our best knowledge, this is the first report on using this type of features in Chinese dependency parsing. Although (McDonald et al., 2005) used the prefix of each word form instead of word form itself as features, character-level features here for Chinese is essentially different from that. As Chinese is basically a character-based written language. Character plays an important role in many means, most characters can be formed as single-character words, and Chinese itself is character-order free rather than word-order free to some extent. In addition, there is often a close connection between the meaning of a Chinese word and its first or last character.

4.2 Parsing using a Beam Search Algorithm

In Table 2, the feature $preact_n$ returns the previous parsing action type, and the subscript n stands for the action order before the current action. These are a group of Markovian features. Without this type of features, a shift-reduce parser may directly scan through an input sequence in linear time. Otherwise, following the work of (Duan et al., 2007) and (Zhao, 2009), the parsing algorithm is to search a parsing action sequence with the maximal probability.

$$S_{d_i} = \text{argmax} \prod_i p(d_i|d_{i-1}d_{i-2}...),$$

where S_{d_i} is the object parsing action sequence, $p(d_i|d_{i-1}...)$ is the conditional probability, and d_i



1	The	the	DT	2	NMOD
2	company	company	NN	3	SBJ
3	said	say	VBD	0	ROOT
4	it	it	PRP	5	SBJ
5	will	will	MD	3	OBJ
6	use	use	VB	5	VC
7	the	the	DT	8	NMOD
8	proceeds	proceeds	NNS	6	OBJ
9	of	of	IN	8	NMOD
10	the	the	DT	11	NMOD
11	offering	offering	NN	9	PMOD
12	for	for	IN	6	ADV
13	debt	debt	NN	14	NMOD
14	reduction	reduction	NN	12	PMOD
15	and	and	CC	14	COORD
16	general	general	JJ	18	NMOD
17	corporate	corporate	JJ	18	NMOD
18	purposes	purpose	NNS	15	CONJ
19	,	,	,	18	P
20	including	include	VBG	18	NMOD
21	acquisitions	acquisition	NNS	20	PMOD
22	.	.	.	3	P

1	企业	NN	16	SBJ
2	共	JJ	21	NMOD
3	获得	NNS	13	PMOD
4	收益	NNS	9	OBJ
5	,	,	21	P
6	它	PRP	7	SBJ
7	会	MD	16	OBJ
8	减少	NN	11	PMOD
9	利用	VB	7	VC
10	这	DT	1	NMOD
11	对	IN	9	ADV
12	这	DT	15	NMOD
13	包括	VBG	21	NMOD
14	团结的	JJ	21	NMOD
15	提供	NN	18	PMOD
16	说	VBD	0	ROOT
17	这	DT	4	NMOD
18	的	IN	4	NMOD
19	债务	NN	8	NMOD
20	及	CC	8	COORD
21	目的	NNS	20	CONJ
22	。	.	16	P

Figure 1: A comparison before and after translation

Table 2: Features for Parsing

_	$i_n.form, n = 0, 1$
_	$i.form + i_1.form$
_	$i_n.char_2 + i_{n+1}.char_2, n = -1, 0$
_	$i.char_{-1} + i_1.char_{-1}$
_	$i_n.char_{-2}\ n = 0, 3$
_	$i_1.char_{-2} + i_2.char_{-2} + i_3.char_{-2}$
_	$i.lnverb.char_{-2}$
_	$i_3.pos$
_	$i_n.pos + i_{n+1}.pos, n = 0, 1$
_	$i_{-2}.cpos1 + i_{-1}.cpos1$
_	$i_1.cpos1 + i_2.cpos1 + i_3.cpos1$
_	$s'_2.char_1$
_	$s'.char_{-2} + s'_1.char_{-2}$
_	$s'_{-2}.cpos2$
_	$s'_{-1}.cpos2 + s'_1.cpos2$
_	$s'.cpos2 + s'_1.cpos2$
_	*s'.children.cpos2.seq*
_	*s'.children.dprel.seq*
_	*s'.subtree.depth*
_	$s'.h.form + s'.rm.cpos1$
_	$s'.lm.char_2 + s'.char_2$
_	*s.h.children.dprel.seq*
_	*s.lm.dprel*
_	$s.char_{-2} + i_1.char_{-2}$
_	$s.char_n + i.char_n, n = -1, 1$
_	$s_{-1}.pos + i_1.pos$
_	$s.pos + i_n.pos, n = -1, 0, 1$
_	$s : i|linePath.form.bag$
_	$s'.form + i.form$
_	$s'.char_2 + i_n.char_2, n = -1, 0, 1$
_	$s.curroot.pos + i.pos$
_	$s.curroot.char_2 + i.char_2$
_	$s.children.cpos2.seq + i.children.cpos2.seq$
_	$s.children.cpos2.seq + i.children.cpos2.seq + s.cpos2 + i.cpos2$
_	$s'.children.dprel.seq + i.children.dprel.seq$
_	$preact_{-1}$
_	$preact_{-2}$
_	$preact_{-2}$+$preact_{-1}$

is i-th parsing action. We use a beam search algorithm to find the object parsing action sequence.

5 Exploiting the Translated Treebank

As we cannot expect too much for a word-by-word translation, only word pairs with dependency relation in translated text are extracted as useful and reliable information. Then some features based on a query in these word pairs according to the current parsing state (namely, words in the current stack and input) will be derived to enhance the Chinese parser.

A translation sample can be seen in Figure 1. Although most words are satisfactorily translated, to generate effective features, what we still have to consider at first is the inconsistence between the translated text and the target text.

In Chinese, word lemma is always its word form itself, this is a convenient characteristic in computational linguistics and makes lemma features unnecessary for Chinese parsing at all. However, Chinese has a special primary processing task, i.e., word segmentation. Unfortunately, word definitions for Chinese are not consistent in various linguistical views, for example, seven segmentation conventions for computational purpose are formally proposed since the first Bakeoff[3].

Note that CTB or any other Chinese treebank has its own word segmentation guideline. Chinese word should be strictly segmented according to the guideline before POS tags and dependency relations are annotated. However, as we say the

[3]Bakeoff is a Chinese processing share task held by SIGHAN.

English treebank is translated into Chinese word by word, Chinese words in the translated text are exactly some entries from the bilingual lexicon, they are actually irregular phrases, short sentences or something else rather than words that follows any existing word segmentation convention. If the bilingual lexicon is not carefully selected or refined according to the treebank where the Chinese parser is trained from, then there will be a serious inconsistence on word segmentation conventions between the translated and the target treebanks.

As all concerned feature values here are calculated from the searching result in the translated word pair list according to the current parsing state, and a complete and exact match cannot be always expected, our solution to the above segmentation issue is using a partial matching strategy based on characters that the words include.

Above all, a translated word pair list, L, is extracted from the translated treebank. Each item in the list consists of three elements, dependant word (dp), head word (hd) and the frequency of this pair in the translated treebank, f.

There are two basic strategies to organize the features derived from the translated word pair list. The first is to find the most matching word pair in the list and extract some properties from it, such as the matched length, part-of-speech tags and so on, to generate features. Note that a matching priority serial should be defined aforehand in this case. The second is to check every matching models between the current parsing state and the partially matched word pair. In an early version of our approach, the former was implemented. However, It is proven to be quite inefficient in computation. Thus we adopt the second strategy at last. Two matching model feature functions, $\phi(\cdot)$ and $\psi(\cdot)$, are correspondingly defined as follows. The return value of $\phi(\cdot)$ or $\psi(\cdot)$ is the logarithmic frequency of the matched item. There are four input parameters required by the function $\phi(\cdot)$. Two parameters of them are about which part of the stack(input) words is chosen, and other two are about which part of each item in the translated word pair is chosen. These parameters could be set to $full$ or $char_n$ as shown in Table 1, where $n = ..., -2, -1, 1, 2, ...$ For example, a possible feature could be $\phi(s.full, i.char_1, dp.full, hd.char_1)$, it tries to find a match in L by comparing stack word and dp word, and the first character of input word and the first character of hd word. If such a match item in L is found, then $\phi(\cdot)$ returns $\log(f)$. There are three input parameters required by the function $\psi(\cdot)$. One parameter is about which part of the stack(input) words is chosen, and the other is about which part of each item in the translated word pair is chosen. The third is about the matching type that may be set to $dependant$, $head$, or $root$. For example, the function $\psi(i.char_1, hd.full, root)$ tries to find a match in L by comparing the first character of input word and the whole dp word. If such a match item in L is found, then $\psi(\cdot)$ returns $\log(f)$ as hd occurs as ROOT f times.

Table 3: Features based on the translated treebank

-	$\phi(i.char_3, s'.full, dp.char_3, hd.full)$+$i.char_3$ +$s'.form$
-	$\phi(i.char_3, s.char_2, dp.char_3, hd.char_2)$+$s.char_2$
-	$\phi(i.char_3, s.full, dp.char_3, hd.char_2)$+$s.form$
-	$\psi(s'.char_{-2}, hd.char_{-2}, head)$+$i.pos$+$s'.pos$
-	$\phi(i.char_3, s.full, dp.char_3, hd.char_2)$+$s.full$
-	$\phi(s'.full, i.char_4, dp.full, hd.char_4)$+$s'.pos$+$i.pos$
-	$\psi(i.full, hd.char_2, root)$+$i.pos$+$s.pos$
-	$\psi(i.full, hd.char_2, root)$+$i.pos$+$s'.pos$
-	$\psi(s.full, dp.full, dependant)$+$i.pos$
-	$pairscore(s'.pos, i.pos)$+$s'.form$+$i.form$
-	$rootscore(s'.pos)$+$s'.form$+$i.form$
-	$rootscore(s'.pos)$+$i.pos$

As having observed that CTB and PTB share a similar POS guideline. A POS pair list from PTB is also extract. Two types of features, $rootscore$ and $pairscore$ are used to make use of such information. Both of them returns the logarithmic value of the frequency for a given dependent event. The difference is, $rootscore$ counts for the given POS tag occurring as ROOT, and $pairscore$ counts for two POS tag combination occurring for a dependent relationship.

A full adapted feature list that is derived from the translated word pairs is in Table 3.

6 Evaluation Results

The quality of the parser is measured by the parsing accuracy or the unlabeled attachment score (UAS), i.e., the percentage of tokens with correct head. Two types of scores are reported for comparison: "UAS without p" is the UAS score without all punctuation tokens and "UAS with p" is the one with all punctuation tokens.

The results with different feature sets are in Table 4. As the features $preact_n$ are involved, a

beam search algorithm with width 5 is used for parsing, otherwise, a simple shift-reduce decoding is used. It is observed that the features derived from the translated text bring a significant performance improvement as high as 1.3%.

Table 4: The results with different feature sets

	features	with p	without p
baseline	-d	0.846	0.858
	+d[a]	0.848	0.860
+T[b]	-d	0.859	0.869
	+d	**0.861**	**0.870**

[a]+d: using three Markovian features *preact* and beam search decoding.

[b]+T: using features derived from the translated text as in Table 3.

To compare our parser to the state-of-the-art counterparts, we use the same testing data as (Wang et al., 2005) did, selecting the sentences length up to 40. Table 5 shows the results achieved by other researchers and ours (UAS with p), which indicates that our parser outperforms any other ones [4]. However, our results is only slightly better than that of (Chen et al., 2008) as only sentences whose lengths are less than 40 are considered. As our full result is much better than the latter, this comparison indicates that our approach improves the performance for those longer sentences.

Table 5: Comparison against the state-of-the-art

	full	up to 40
(McDonald and Pereira, 2006)[a]	-	0.825
(Wang et al., 2007)	-	0.866
(Chen et al., 2008)	0.852	0.884
Ours	**0.861**	**0.889**

[a]This results was reported in (Wang et al., 2007).

The experimental results in (McDonald and Nivre, 2007) show a negative impact on the parsing accuracy from too long dependency relation. For the proposed method, the improvement relative to dependency length is shown in Figure 2. From the figure, it is seen that our method gives observable better performance when dependency lengths are larger than 4. Although word order is changed, the results here show that the useful information from the translated treebank still help those long distance dependencies.

[4]There is a slight exception: using the same data splitting, (Yu et al., 2008) reported UAS without p as 0.873 versus ours, 0.870.

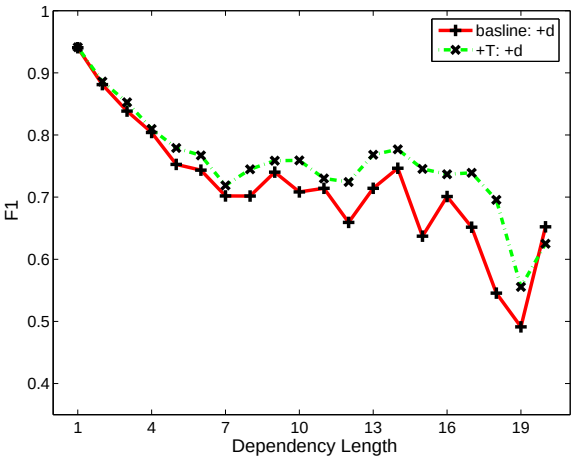


Figure 2: Performance vs. dependency length

7 Discussion

If a treebank in the source language can help improve parsing in the target language, then there must be something common between these two languages, or more precisely, these two corresponding treebanks. (Zeman and Resnik, 2008) assumed that the morphology and syntax in the language pair should be very similar, and that is so for the language pair that they considered, Danish and Swedish, two very close north European languages. Thus it is somewhat surprising that we show a translated English treebank may help Chinese parsing, as English and Chinese even belong to two different language systems. However, it will not be so strange if we recognize that PTB and CTB share very similar guidelines on POS and syntactics annotation. Since it will be too abstract in discussing the details of the annotation guidelines, we look into the similarities of two treebanks from the matching degree of two word pair lists. The reason is that the effectiveness of the proposed method actually relies on how many word pairs at every parsing states can find their full or partial matched partners in the translated word pair list. Table 6 shows such a statistics on the matching degree distribution from all training samples for Chinese parsing. The statistics in the table suggest that most to-be-check word pairs during parsing have a full or partial hitting in the translated word pair list. The latter then obtains an opportunity to provide a great deal of useful guideline information to help determine how the former should be tackled. Therefore we have cause for attributing the effectiveness of the proposed method to the similarity of these two treebanks. From Table 6,

we also find that the partial matching strategy defined in Section 5 plays a very important role in improving the whole matching degree. Note that our approach is not too related to the characteristics of two languages. Our discussion here brings an interesting issue, which difference is more important in cross language processing, between two languages themselves or the corresponding annotated corpora? This may be extensively discussed in the future work.

Table 6: Matching degree distribution

dependant-match	head-match	Percent (%)
None	None	9.6
None	Partial	16.2
None	Full	9.9
Partial	None	12.4
Partial	Partial	42.6
Partial	Full	7.3
Full	None	3.7
Full	Partial	7.0
Full	Full	0.2

Note that only a bilingual lexicon is adopted in our approach. We regard it one of the most merits for our approach. A lexicon is much easier to be obtained than an annotated corpus. One of the remained question about this work is if the bilingual lexicon should be very specific for this kind of tasks. According to our experiences, actually, it is not so sensitive to choose a highly refined lexicon or not. We once found many words, mostly named entities, were outside the lexicon. Thus we managed to collect a named entity translation dictionary to enhance the original one. However, this extra effort did not receive an observable performance improvement in return. Finally we realize that a lexicon that can guarantee two word pair lists highly matched is sufficient for this work, and this requirement may be conveniently satisfied only if the lexicon consists of adequate high-frequent words from the source treebank.

8 Conclusion and Future Work

We propose a method to enhance dependency parsing in one language by using a translated treebank from another language. A simple statistical machine translation technique, word-by-word decoding, where only a bilingual lexicon is necessary, is used to translate the source treebank. As dependency parsing is concerned with the relations of word pairs, only those word pairs with dependency relations in the translated treebank are chosen to generate some additional features to enhance the parser for the target language. The experimental results in English and Chinese treebanks show the proposed method is effective and helps the Chinese parser in this work achieve a state-of-the-art result.

Note that our method is evaluated in two treebanks with a similar annotation style and it avoids using too many linguistic properties. Thus the method is in the hope of being used in other similarly annotated treebanks [5]. For an immediate example, we may adopt a translated Chinese treebank to improve English parsing. Although there are still something to do, the remained key work has been as simple as considering how to determine the matching strategy for searching the translated word pair list in English according to the framework of our method. .

Acknowledgements

We'd like to give our thanks to three anonymous reviewers for their insightful comments, Dr. Chen Wenliang for for helpful discussions and Mr. Liu Jun for helping us fix a bug in our scoring program.

References

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *EMNLP-2008*, pages 877–886, Honolulu, Hawaii, USA.

Wenliang Chen, Daisuke Kawahara, Kiyotaka Uchimoto, Yujie Zhang, and Hitoshi Isahara. 2008. Dependency parsing with short dependency relations in unlabeled data. In *Proceedings of IJCNLP-2008*, Hyderabad, India, January 8-10.

Xiangyu Duan, Jun Zhao, and Bo Xu. 2007. Probabilistic parsing action models for multi-lingual dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 940–946, Prague, Czech, June 28-30.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryiğit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or

[5]For example, Catalan and Spanish treebanks from the AnCora(-Es/Ca) Multilevel Annotated Corpus that are annotated by the Universitat de Barcelona (CLiC-UB) and the Universitat Politècnica de Catalunya (UPC).

blended? a study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 933–939, Prague, Czech, June.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, Ohio, USA, June.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of ACL-COLING 2006*, pages 337–344, Sydney, Australia, July.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 122–131, Prague, Czech, June 28-30.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL-2006*, pages 81–88, Trento, Italy, April.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL-2005*, pages 91–98, Ann Arbor, Michigan, USA, June 25-30.

Paola Merlo, Suzanne Stevenson, Vivian Tsang, and Gianluca Allaria. 2002. A multilingual paradigm for automatic verb classification. In *ACL-2002*, pages 207–214, Philadelphia, Pennsylvania, USA.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, page 915 - 932, Prague, Czech, June.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT-2003)*, pages 149–160, Nancy, France, April 23-25.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL-2002*, pages 295–302, Philadelphia, USA, July.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of ACL-2007*, pages 616–623, Prague, Czech Republic, June.

Kenji Sagae and Jun' ichi Tsujii. 2007. Dependency parsing and domain adaptation with lr models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, page 1044 - 1050, Prague, Czech, June 28-30.

Noah A. Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of ACL-COLING 2006*, page 569 - 576, Sydney, Australia, July.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of EACL-2003*, page 331 - 338, Budapest, Hungary, April.

Qin Iris Wang and Dale Schuurmans. 2008. Semi-supervised convex training for dependency parsing. In *Proceedings of ACL-08: HLT*, pages 532–540, Columbus, Ohio, USA, June.

Qin Iris Wang, Dale Schuurmans, and Dekang Lin. 2005. Strictly lexical dependency parsing. In *Proceedings of IWPT-2005*, pages 152–159, Vancouver, BC, Canada, October.

Qin Iris Wang, Dekang Lin, and Dale Schuurmans. 2007. Simple training of dependency parsers via structured boosting. In *Proceedings of IJCAI 2007*, pages 1756–1762, Hyderabad, India, January.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT-2003)*, page 195 - 206, Nancy, France, April.

Kun Yu, Daisuke Kawahara, and Sadao Kurohashi. 2008. Chinese dependency parsing with large scale automatically constructed case structures. In *Proceedings of COLING-2008*, pages 1049–1056, Manchester, UK, August.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of IJCNLP 2008 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India, January.

Hai Zhao and Chunyu Kit. 2008. Parsing syntactic and semantic dependencies with two single-stage maximum entropy models. In *Proceeding of CoNLL-2008*, pages 203–207, Manchester, UK.

Hai Zhao, Wenliang Chen, Chunyu Kit, and Guodong Zhou. 2009. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of CoNLL-2009*, Boulder, Colorado, USA.

Hai Zhao. 2009. Character-level dependencies in chinese: Usefulness and learning. In *EACL-2009*, pages 879–887, Athens, Greece.