Accurate Learning for Chinese Function Tags from Minimal Features

Caixia Yuan[1,2], **Fuji Ren**[1,2] **and Xiaojie Wang**[2]
[1]The University of Tokushima, Tokushima, Japan
[2]Beijing University of Posts and Telecommunications, Beijing, China
{yuancai,ren}@is.tokushima-u.ac.jp
xjwang@bupt.edu.cn

Abstract

Data-driven function tag assignment has been studied for English using Penn Treebank data. In this paper, we address the question of whether such method can be applied to other languages and Treebank resources. In addition to simply extend previous method from English to Chinese, we also proposed an effective way to recognize function tags directly from lexical information, which is easily scalable for languages that lack sufficient parsing resources or have inherent linguistic challenges for parsing. We investigated a supervised sequence learning method to automatically recognize function tags, which achieves an F-score of 0.938 on gold-standard POS (Part-of-Speech) tagged Chinese text – a statistically significant improvement over existing Chinese function label assignment systems. Results show that a small number of linguistically motivated lexical features are sufficient to achieve comparable performance to systems using sophisticated parse trees.

1 Introduction

Function tags, such as subject, object, time, location, etc. are conceptually appealing by encoding an event in the format of "who did what to whom, where, when", which provides useful semantic information of the sentences. Lexical semantic resources such as Penn Treebank (Marcus et al., 1994) have been annotated with phrase tree structures and function tags. Figure 1 shows the parse tree with function tags for a sample sentence form the Penn Chinese Treebank 5.0[1] (Xue et al., 2000) (file 0043.fid).

[1]released by Linguistic Data Consortium (LDC) catalog NO. LDC2005T01

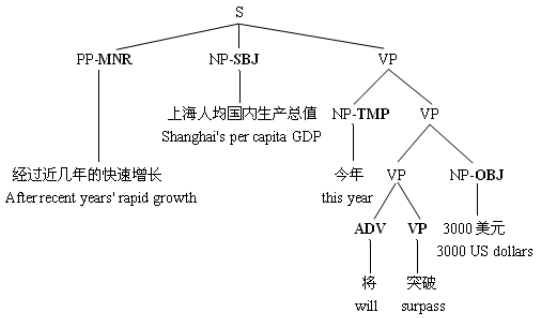


Figure 1: Simplified parse tree with function tags (in black bold) for example sentence.

When dealing with the task of function tag assignment (or function labeling thereafter), one basic question that must be addressed is what features can be extracted in practice for distinguishing different function tag types. In answering this question, several pieces of work (Blaheta and Charniak, 2000; Blaheta, 2004; Merlo and Musillo, 2005; Gildea and Palmer, 2002) have already been proposed. (Blaheta and Charniak, 2000; Blaheta, 2004) described a statistical system trained on the data of Penn Treebank to automatically assign function tags for English text. The system first passed sentences through an automatic parser, then extracted features from the parse trees and predicted the most plausible function label of constituent from these features. Noting that parsing errors are difficult or even impossible to recover at function tag recognition stage, the alternative approaches are obtained by assigning function tags at the same time as producing parse trees (Merlo and Musillo, 2005), through learning deeper syntactic properties such as finer-grained labels, features from the nodes to the left of the current node.

Through all that research, however, successfully addressing function labeling requires accurate parsing model and training data, and the re-

Proceedings of the ACL-IJCNLP 2009 Student Research Workshop, pages 54–62,
Suntec, Singapore, 4 August 2009.

sults of them show that the performance ceiling of function labeling is limited by the parsers they used. Given the imperfection of existing automatic parsers, which are far from producing gold-standard results, function tags output by such models cannot be satisfactory for practical use. The limitation is even more pertinent for the languages that do not have sophisticated parsing resources, or languages that have inherent linguistic challenges for parsing (like Chinese). It is therefore worthwhile to investigate alternatives to function labeling for languages under the parsing bottleneck, both in terms of features used and effective learning algorithms.

In current study, we focused on the use of parser-independent features for function labeling. Specifically, our proposal is to classify function types directly from lexical features like words and their POS tags and the surface sentence information like the word position. The hypothesis that underlies our proposal is that lexical features are informative for different function types, and capture fundamental properties of the semantics that sometimes can not be concluded from the glance of parse structure. Such cases come when distinguishing phrases of the same structure that differ by just one word – for instance, telling "在上海 (in Shanghai)", which is locative, from "在五月 (in May)", which is temporal.

At a high level, we can say that class-based differences in function labels are reflected in statistics over the lexical features in large-scale annotated corpus, and that such knowledge can be encoded by learning algorithms. By exploiting lexical information collected from Penn Chinese Treebank (CTB) (Xue et al., 2000), we investigate a supervised sequence learning model to test our core hypothesis – that function tags could be guessed precisely through informative lexical features and effective learning methods. At the end of this paper, we extend previous function labeling methods from English to Chinese. The result proves, at least for Chinese language, our proposed method outperforms previous ones that utilize sophisticated parse trees.

In section 2 we will introduce the CTB resources and function tags used in our study. In section 3, we will describe the sequence learning algorithm in the framework of maximum margin learning, showing how to approximate function tagging by simple lexical statistics. Section 4 gives a detailed discussion of our experiment and comparison with pieces of related work. Some final remarks will be given in Section 5.

Table 1: Complete set of function labels in Chinese Treebank and function labels used in our system (selected labels).

type	labels in CTB		selected labels
clause types	IMP	imperative	
	Q	question	
(function/form) discrepancies	ADV	adverbial	√
grammatical roles	EXT	extent	√
	FOC	focus	√
	IO	indirect object	√
	OBJ	direct object	√
	PRD	predicate	√
	SBJ	subject	√
	TPC	topic	√
adverbials	BNF	beneficiary	√
	CND	condition	√
	DIR	direction	√
	IJ	interjective	√
	LGS	logic subject	√
	LOC	locative	√
	MNR	manner	√
	PRP	purpose/reason	√
	TMP	temporal	√
	VOC	vocative	√
miscellaneous	APP	appositive	
	HLN	headline	
	PN	proper names	
	SHORT	short form	
	TTL	title	
	WH	wh-phrase	

2 Chinese Function Tags

The label such as subject, object, time, location, etc. are named as function tags[2] in Penn Chinese Treebank (Xue et al., 2000), a complete list of which is shown in Table 1. Among the 5 categories, grammatical roles such as SBJ, OBJ are useful in recovering predicate-argument structure, while adverbials are actually semantically oriented labels (though not true for all cases, see (Merlo and Palmer, 2006)) that carry semantic role information.

As for the task of function parsing, it is reasonable to ignore the IMP and Q in Table 1 since they do not form natural syntactic or semantic classes. In addition, we regard the miscellaneous labels as an "O" label (out of any function chunks) like labeling constituents that do not bear any function

[2]The annotation guidelines of Penn Chinese Treebank talk of function tags. We will use the term function labels and function tags identically, and hence make no distinction between function labeling and function tagging throughout this paper. Also, the term function chunk signifies a sequence of words that are decorated with the same function label.

tags. Punctuation marks like comma, semi-colon and period that separate sentences are also denoted as "O". But the punctuation that appear within one sentence like double quotes are denoted with the same function labels with the content they quote.

In the annotation guidelines of CTB (Xue et al., 2000), the function tag "PRD" is assigned to non-verbal predicate. Since VP (verb phrase) is always predicate, "PRD" is assumed and no function tag is attached to it. We make a slight modification to such standard by calling this kind of VP "verbal predicates", and assigning them with function label "TAR (target verb)", which is grouped into the same grammar roles type with "PRD".

To a large extent, PP (preposition phrase) always plays a functional role in sentence, like "PP-MNR" in Figure 1. But there are many such PPs bare of any function type in CTB resources. Like in the sentence "比去年同期增长 25% (increase by 25% over the same period of last year)", "比去年同期 (over the same period of last year)" is labeled as "PP" in CTB without any function labels attached, thus losing to describe the relationship with the predicate "增长 (increases)". In order to capture various relationships related to the predicate, we assign function label "ADT (adjunct)" for this scenario, and merge it with other adverbials to form adverbials category. There are 1,415 such cases in CTB resources, which account for a large proportion of adverbials types.

After the modifications discussed above, in our final system we use 20 function labels[3] (18 original CTB labels shown in Table 2 and two newly added labels) that are grouped into two types: grammatical roles and adverbials.

We calculate the frequency (the number of times each tag occurs) and average length (the average number of words each tag covers) of each function category in our selected sentences, which are listed in Table 2. As can be seen, the frequency of adverbials is much smaller than that of grammatical roles. Furthermore, the average length of most adverbials are somewhat larger than 4. Such data distribution is likely to be one cause of the lower identification accuracy of adverbials as we will see in the experiments.

From the layer of function labeling, sentences in CTB are described with the structure of "SV" which indicates a sentence is basically composed of "subject + verb". But in order to identify objects and complements of predicates, we express sentence by "SVO" framework in our system, which regards sentence as a structure of "subject + verb + object". The structure transformation is obtained through a preprocessing procedure, by upgrading OBJs and complements (EXT, DIR, etc.) which are under VP in layered brackets.

[3]ADV includes ADV and ADVP in CTB recourses, grouped into adverbials. In function labeling level, EXT that signifies degree, amount of the predicates should be grouped into adverbials like in the work of (Blaheta and Charniak, 2000) and (Merlo and Musillo, 2005).

Table 2: Categories of function tags with their relative frequencies and average length.

Function Labels	Frequency	Average Length
grammatical roles	99507	2.62
FOC	133	1.89
IO	126	1.26
OBJ	25834	4.15
PRD	4428	5.20
SBJ	23809	3.02
TPC	676	3.51
TAR	44501	1.25
adverbials	33287	2.11
ADT	1415	4.51
ADV	21891	1.32
BNF	465	4.66
CND	68	3.15
DIR	1558	4.68
EXT	1048	1.99
IJ	1	1.00
LGS	204	5.42
LOC	2051	4.27
MNR	1053	4.48
PRP	224	4.91
TMP	3309	2.25

3 Learning Function Labels

Function labeling deals with the problem of predicting a sequence of function tags $y = y_1, ..., y_T$, from a given sequence of input words $x = x_1, ..., x_T$, where $y_i \in \Sigma$. Therefore the function labeling task can be formulated as a stream of sequence learning problem. The general approach is to learn a w-parameterized mapping function $F : X \times Y \rightarrow \Re$ based on training sample of input-output pairs and to maximize $F(x, y; w)$ over the response variable to make a prediction.

There has been several algorithms for labeling sequence data including hidden Markov model (Rabiner, 1989), maximum entropy Markov model (Mccallum et al., 2000), conditional random fields (Lafferty et al., 2001) and hidden Markov support vector machine (HM-SVM) (Altun et al., 2003; Tsochantaridis et al., 2004), among which HM-SVM shows notable advantages by its learning

non-linear discriminant functions via kernel function, the properties inherited from support vector machines (SVMs). Furthermore, HM-SVM retains some of the key advantages of Markov model, namely the Markov chain dependency structure between labels and an efficient dynamic programming formulation.

In this paper we investigate the application of the HM-SVM model to Chinese function labeling task. In order to keep the completeness of paper, we here address briefly the HM-SVM algorithm, more details of which could be founded in (Altun et al., 2003; Tsochantaridis et al., 2004), then we will concentrate on the techniques of applying it to our specific task.

3.1 Learning Model

The framework from which HM-SVM are derived is a maximum margin formulation for joint feature functions in kernel learning setting. Given n labeled examples $(x^1, y^1), ..., (x^n, y^n)$, the notion of a separation margin proposed in standard SVMs is generalized by defining the margin of a training example with respect to a discriminant function $F(x, y; w)$, as:

$$\gamma_i = F(x^i, y^i; w) - \max_{y \notin y^i} F(x^i, y; w). \quad (1)$$

Then the maximum margin problem can be defined as finding a weight vector w that maximizes $min_i \gamma_i$. By fixing the functional margin ($max_i \gamma_i \geq 1$) like in the standard setting of SVMs with binary labels, we get the following hard-margin optimization problem with a quadratic objective:

$$\min_w \frac{1}{2}||w||^2, \quad (2)$$

with constraints,

$$F(x^i, y^i; w) - F(x^i, y; w) \geq 1, \forall_{i=1}^n, \forall_{y \neq y^i}.$$

In the particular setting of SVM, F is assumed to be linear in some combined feature representation of inputs and outputs $\Phi(x, y)$, i.e. $F(x, y; w) = \langle w, \Phi(x, y) \rangle$. $\Phi(x, y)$ can be specified by extracting features from an observation/label sequence pair (x, y). Inspired by HMMs, we propose to define two types of features, interactions between neighboring labels along the chain as well as interactions between attributes of the observation vectors and a specific label. For instance, in our function labeling task, we might think of a label-label feature of the form

$$\alpha(y_{t-1}, y_t) = [[y_{t-1} = \text{SBJ} \wedge y_t = \text{TAR}]], \quad (3)$$

that equals 1 if a SBJ is followed by a TAR. Analogously, a label-observation feature may be

$$\beta(x_t, y_t) = [[y_t = \text{SBJ} \wedge x_t \text{ is a noun}]], \quad (4)$$

which equals 1 if x at position t is a noun and labeled as SBJ. The described feature map exhibits a first-order Markov property and as a result, decoding can be performed by a Viterbi algorithm in $O(T|\Sigma|^2)$.

All the features extracted at location t are simply stacked together to form $\Phi(x, y; t)$. Finally, this feature map is extended to sequences (x, y) of length T in an additive manner as

$$\Phi(x, y) = \sum_{t=1}^{T} \Phi(x, y; t). \quad (5)$$

3.2 Features

It deserves to note that features in HM-SVM model can be easily changeable regardless of dependency among them. In this prospect, features are very far from independent can be cooperated in the model.

By observing the particular property of function structure in Chinese sentences, we design several sets of label-observation features which are independent of parse trees, namely:

Words and POS tags: The lexical context is extremely important in function labeling, as indicated by their importance in related task of phrase chunking. Due to long-distance dependency of function structure, intuitively, more wider context window will bring more accurate prediction. However, the wider context window is more likely to bring sparseness problem of features and increase computation cost. So there should be a proper compromise among them. In our experiment, we start from a context of [-2, +2] and then expand it to [-4, 4], that is, four words (and POS tags) around the word in question, which is closest to the average length of most function types shown in Table 2.

Bi-gram of POS tags: Apart from POS tags themselves, we also try on the bi-gram of POS tags. We regard POS tag sequence as an analog to function

chains, which reveals somewhat the dependent relations among words.

Verbs: Function labels like subject and object specify the relations between verb and its arguments. As observed in English verbs (Levin, 1993), each class of verb is associated with a set of syntactic frames. Similar criteria can also be found in Chinese. In this sense, we can rely on the surface verb for distinguishing argument roles syntactically. Besides the verbs themselves, we also take into account the special words sharing common property with verbs in Chinese language, which are active voice "把(BA)" and passive voice "被(BEI)". The verb we refer here is supposed to be the last verb if it happens in a consecutive verb sequence, thus actually not the head verb of sentence.

POS tags of verbs: according to CTB annotation guideline, verbs are labeled with four kinds of POS tags (VA, VC, VE, VV), along with BA (for "把"), LB and SB (for "被"). This feature somewhat notifies the coarse class of verbs talked in (Levin, 1993) and is taken into account as feature candidates.

Position indicators: It is interesting to notice that whether the constituent to be labeled occurs before or after the verb is highly correlated with grammatical function, since subjects will generally appear before a verb, and objects after, at least for Chinese language. This feature may overcome the lack of syntactic structure that could be read from the parse tree.

In our experiment, all feature candidates are introduced to the training instances incrementally by a feature inducing procedure, then we use a gain-driven method to decide whether a feature should be reserved or deleted according to the increase or decrease of the predication accuracy. The procedure are described in Figure 2.

Figure 2: Pseudo-code of feature introducing procedure.

1: initialize feature superset C={all feature candidates}, feature set c is empty
2: **repeat**
3: **for** each feature $c_i \in C$ **do**
4: construct training instances using $c_i \cup c$
experiment on k-fold cross-validation data
5: **if** accuracy increases **then**
$c_i \rightarrow c$
6: **end if**
7: **end for**
8: **until** all features in C are traversed

4 Experiment and Discussion

In this section, we turn to our computational experiments that investigate whether the statistical indicators of lexical properties that we have developed can in fact be used to classify function labels, and demonstrate which kind of feature contributes most in identifying function types, at least for Chinese text.

As in the work of (Ramshaw and Marcus, 1995), each word or punctuation mark within a sentence is labeled with "IOB" tag together with its function type. The three tags are sufficient for encoding all constituents since there are no overlaps among different function chunks. The function tags in this paper are limited to 20 types, resulting in a total of $|\Sigma| = 41$ different outputs.

We use three measures to evaluate the model performance: *precision*, which is the percentage of detected chunks that are correct; *recall*, which is the percentage of chunks in the data that are found by the tagger; and *F-score* which is equal to $2 \times precision \times recall/(precision + recall)$. Under the "IOB" tagging scheme, a function chunk is only counted as correct when its boundaries and its type are both identified correctly. Furthermore, sentence accuracy is used in order to observe the prediction correctness of sentences, which is defined as the percentage of sentences within which all the constituents are assigned with correct tags. As in the work of (Blaheta and Charniak, 2000) and (Merlo and Musillo, 2005), to avoid calculating excessively optimistic values, constituents bearing the "O" label are not counted in for computing overall precision, recall and F-score.

We derived 18,782 sentences from CTB 5.0 with about 497 thousands of words (including punctuation marks). On average, each sentence contains 26.5 words with 2.4 verbs. We followed 5-fold cross-validation method in our experiment. The numbers reported are the averages of the results across the five test sets.

4.1 Evaluation of Different Features and Models

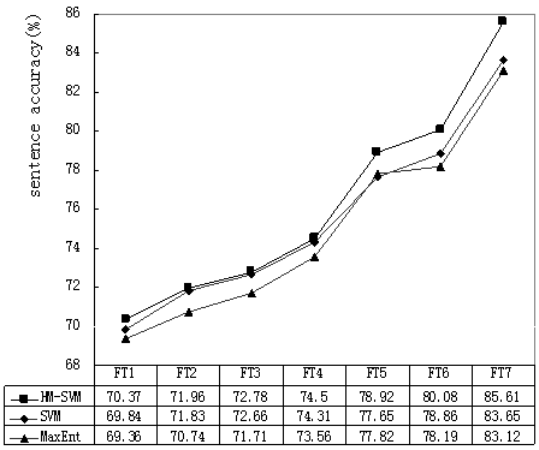
In pilot experiments on a subset of the features, we provide a comparison of HM-SVM with other two learning models, maximum entropy (MaxEnt) model (Berger et al., 1996) and SVM model (Kudo, 2001), to test the effectiveness of HM-SVM on function labeling task, as well as the generality of our hypothesis on different learning

Table 3: Features used in each experiment round.

FT1	word & POS tags within [-2,+2]
FT2	word & POS tags within [-3,+3]
FT3	word & POS tags within [-4,+4]
FT4	FT3 plus POS bigrams within [-4,+4]
FT5	FT4 plus verbs
FT6	FT5 plus POS tags of verbs
FT7	FT6 plus position indicators

models.

In our experiment, SVMs and HM-SVM training are carried out with SVM^{struct} packages[4]. The multi-class SVMs model is realized by extending binary SVMs using *pairwise* strategy. We used a first-order of transition and emission dependency in HM-SVM. Both SVMs and HM-SVM are trained with the linear kernel function and the soft margin parameter c is set to be 1. The MaxEnt model is implemented based on Zhang's MaxEnt toolkit[5] and L-BFGS (Nocedal, 1999) method to perform parameter estimation.



	FT1	FT2	FT3	FT4	FT5	FT6	FT7
HM-SVM	70.37	71.96	72.78	74.5	78.92	80.08	85.61
SVM	69.84	71.83	72.66	74.31	77.65	78.86	83.65
MaxEnt	69.36	70.74	71.71	73.56	77.82	78.19	83.12

Figure 3: Sentence accuracy achieved by different models using different feature combinations.

We use sentence accuracy to compare performances of three models with different feature combinations shown in Table 3. The learning curves in Figure 3 illustrate feature combination FT7 gains the best results for all three models we considered. As we have expected, the performance improves as the context window expanded from 2 to 4 (from FT1 to FT3 in Figure 3). The sentence accuracy increases significantly when the features include verbs and position indicators, giving some indication of the complexity of the structure intervening between focus word and the verb. However, at a high level, we can simply say that any further information would help for identifying function types, so we believe that the features we deliberated on currently are by no means the solely optimal feature set.

As observed in Figure 3, the structural sequence model HM-SVM outperforms multi-class SVMs, meanwhile, they both perform slightly better than MaxEnt model, demonstrating the benefit of maximum margin based approach. In the experiment below, we will use feature FT7 and HM-SVM model to illustrate our method.

4.2 Results with Gold-standard POS Tags

By using gold-standard POS tags, this experiment is to view the performance of two types of function labels - grammatical roles and adverbials, and fine-grained function types belonging to them. We cite the average precision, recall and F-score of 5-fold cross validation data output by HM-SVM model to discuss this facet.

Table 4: Average performance for individual categories, using HM-SVM model with feature FT7 and gold-standard POS tags.

	Precision	Recall	F-score
Overall	0.934	0.942	0.938
grammatical roles	0.949	0.960	0.955
FOC	0.385	0.185	0.250
IO	0.857	0.286	0.429
OBJ	0.960	0.980	0.970
PRD	0.985	0.988	0.987
SBJ	0.869	0.912	0.890
TPC	0.292	0.051	0.087
TAR	0.986	0.990	0.990
adverbials	0.887	0.887	0.887
ADT	0.690	0.663	0.676
ADV	0.956	0.955	0.956
BNF	0.729	0.869	0.793
CND	0.000	0.000	0.000
DIR	0.741	0.812	0.775
EXT	0.899	0.820	0.857
LGS	0.563	0.659	0.607
LOC	0.712	0.721	0.716
MNR	0.736	0.783	0.759
PRP	0.656	0.404	0.500
TMP	0.821	0.808	0.814

Table 4 details the results of individual function types. On the whole, grammatical roles outperform adverbials. It seems to reflect the fact that

[4]http://svmlight.joachims.org/s vm_multiclass.html

[5]http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

syntactic constituents can often be guessed based on POS tags and high-frequency lexical words, largely avoiding sparse-data problems. This is evident particularly for "OBJ" that reaches aggressively 0.970 in F-score. One exception is "TPC", whose precision and recall draws to the lowest among grammatical roles. In CTB resources, "TPC" marks elements that appear before the subject in a declarative sentence, and, it always constitutes a noun phrase together with the subject of the sentence. As an illustrating example, in the sentence "天津与台湾产业结果相似 (The industrial structure of Tianjin and Taiwan is similar)", "天津与台湾 (Tianjin and Taiwan)" is labeled with "TPC", while "产业结构 (The industrial structure)" with "SBJ". In such settings, it is difficult to distinguish between them even for human beings.

Overall, there are three possible explanations for the lower F-score of adverbials. One is that tags characterized by much more semantic information always have flexible syntactic constructions and diverse positions in sentence, which makes it difficult to capture their uniform characteristics. Second one is likely that the long-distance dependency and sparseness problem degrade the performance of adverbials greatly. This can be viewed from the statistics in Table 2, where most of the adverbials are longer than 4, while the frequency of them is significantly lower than that of grammatical roles. The third possible explanation is that there is vagueness among different adverbials. An instance to state such case is the dispute between "ADV" and "MNR" like the phrase "随着改革开放的深入 (with the deepening of reform and opening-up)", which are assigned with "ADV" and "MNR" in two totally the same contexts in our training data. Noting that word sequences for some semantic labels carry several limited formations (e.g., most of "DIR" is preposition phrase beginning with "from, to"), we will try some linguistically informed heuristics to detect such patterns in future work.

4.3 Results with Automatically Assigned POS Tags

Parallel to experiments on text with gold-standard POS tags, we also present results on automatically POS-tagged text to quantify the effect of POS accuracy on the system performance. We adopt automatic POS tagger of (Qin et al., 2008), which got the first place in the forth SIGHAN Chinese POS tagging bakeoff on CTB open test, to assign POS tags for our data. Following the approach of (Qin et al., 2008), we train the automatic POS tagger which gets an average accuracy of 96.18% in our 5-fold cross-validation data. Function tagger takes raw text as input, then completes POS tagging and function labeling in a cascaded way. As shown in Table 5, the F-score of AutoPOS is slightly lower than that of GoldPOS. However, the small gap is still within our first expectation.

Table 5: Performance separated for grammatical roles and adverbials, of our models GoldPOS (using gold-standard POS tags), GoldPARSE (using gold-standard parse trees), AutoPOS (using automatically labeled POS tags).

	grammatical roles			adverbials		
	P	R	F	P	R	F
GoldPOS	0.949	0.960	**0.955**	0.887	0.887	0.887
AutoPOS	0.921	0.948	0.934	0.872	0.867	0.869
GoldPARSE	0.936	0.967	0.951	0.911	0.884	**0.897**

4.4 Results with Gold-standard Parser

A thoroughly different way for function labeling is deriving function labels together with parsing. The work of (Blaheta and Charniak, 2000; Blaheta, 2004; Merlo and Musillo, 2005) has approved its effectiveness in English text. Among them, the work of Merlo and Musillo (Merlo and Musillo, 2005) achieved a state-of-the-art F1 score for English function labeling (0.964 for grammatical roles and 0.863 for adverbials). In order to address the question of whether such method can be successfully applied to Chinese text and whether the simple method we proposed is better than or at least equivalent to it, we used features collected from hand-crafted parse trees in CTB resources, and did a separate experiment on the same text. The features we used are borrowed from feature trees described in (Blaheta and Charniak, 2000). A trivial difference is that in our system the head for prepositional phrases is defined as the prepositions themselves (not the head of object of prepositional phrases (Blaheta and Charniak, 2000)), because we think that the preposition itself is a more distinctive attribute for different semantic meanings.

Results in Table 5 show that the parser tree doesn't help a lot in Chinese function labeling. One reason for this may be sparseness problem of parse tree features – For instance, in one of the 5-

fold data, 34% of syntactic paths in test instances are unseen in training data. For sentences with the average length of more than 40 words, this sparseness becomes even severe. Another possible reason is that some functional chunks are more local and less prone to structured parse trees, as observed in examples listed at the beginning of the paper. In Table 5, although the performance of adverbials grows really huge when using features from the gold-standard parse trees, the performance of grammatical roles drops as introducing such features. As mentioned above, in fact even the simple position feature can give a better explanation to word's grammatical role than complicated syntactic path.

Although the experimental setup is strictly not the same for the present paper and (Blaheta and Charniak, 2000; Blaheta, 2004; Merlo and Musillo, 2005), we observe that the proposed method yields better results with deliberately designed but simple features at lexical level, while attempts in (Blaheta and Charniak, 2000; Blaheta, 2004; Merlo and Musillo, 2005) optimized function labeling together with parsing, which is a more complex task and difficult to realize for languages that lack sufficient parse resources.

The work of (Blaheta and Charniak, 2000; Blaheta, 2004; Merlo and Musillo, 2005) reveal that the performance of parser used sets upper bound on the performance of function labeling. However, the best Chinese parser ever reported (Wang et al., 2006) achieves 0.882 F-score for sentences with less than 40 words, we therefore conclude that the way using auto-parser for Chinese function labeling is not the optimal choice.

4.5 Error Analysis

In the course of our experiment, we wanted to attain some understanding of what sort of errors the system was making. While still working on the gold-standard POS-tagged text, we randomly took one output from the 5-fold cross-validation tests and examined each error. But when observing the 1,550 wrongly labeled function chunks (26,593 in total), we can distinguish three types of errors.

The first and widest category of errors are caused when the lexical construction of the chunk is similar to other chunk types. A typical example is "PRP (purpose)" and "BNF (beneficiary)", both of which are mostly prepositional phrases beginning with "为, 为了(for, in order to)".

The second type of errors are found when the chunk is too long, like more than 8 words. Normally it is not easy to eliminate this kind of errors through local lexical features. In Chinese, the long chunks are mainly composed of "的 (DE)" structure that can be translated into attributive clause in English. The "的 (DE)" structures are usually nested component and used as a modifier of noun phrases, thus this kind of errors can be partly resolved by accurately recognition of such structure.

The third type of errors concern the sentence with some special structure, like intransitive sentence, elliptical sentence (left out of subject or object), and so on. The errors of "IO" with wrong tag "OBJ", and errors of "EXT" with wrong tag "OBJ" fall into the third categories. It is interesting to notice that, when using GoldPARSE (see Table 5), suggesting that features from the trees are helpful when disambiguating function labels that related with sentence structures.

5 Conclusion and Future Work

We have presented the first experimental results on Chinese function labeling using Chinese Treebank resources, and shown that Chinese function labeling can be reached with considerable accuracy given a small number of lexical features. Even though our experiments using hand-crafted parse trees yield promising initial results, this method will be hampered when using fully automatic parser due to the imperfection of Chinese parser, which is our core motivation to assign function labels by exploiting the underlining lexical insights instead of parse trees. Experimental results suggest that our method for Chinese function labeling is comparable with the English state-of-the-art work that utilizes complicated parse trees.

We believe that we have not settled on an "optimal" set of features for Chinese function labeling, hence, more language-specific customization is necessary in the future work. Although there have been speculations and trails on things that function labels might help with, it remains to be important to discover how function labels contribute to other NLP applications, such as the Japanese-Chinese machine translation system we have been working on.

References

Altun, Y., Tsochantaridis, I., Hofmann, T. 2003. Hidden Markov Support Vector Machines. In: *Pro-*

ceedings of ICML 2003, pages 172-188, Washington, DC, USA.

Berger, A., Pietra, D. S., Pietra, D. V. 1996. A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics, 22(1):39-71.

Blaheta, D. 2004. Function Tagging. Ph.D. thesis, Department of Computer Science, Brown University.

Blaheta, D., Charniak, E. 2000. Assigning Function Tags to Parsed Text. In: *Proceedings of the 1st NAACL*, pages 234-240, Seattle, Washington.

Chrupala, G., Stroppa, N., Genabith, J., Dinu, G. 2007. Better Training for Function Labeling. In: *Proceedings of RANLP2007*, Borovets, Bulgaria.

Gildea, D., Palmer, M. 2002. The Necessity of Parsing for Predicate Argument Recognition. In: *Proceedings of the 40th ACL*, pages 239-246, Philadelphia, USA.

Iida, R., Komachi, M., Inui, K., Matsumoto, Y. 2007. Annotating a Japanese Text Corpus with Predicate-argument and Coreference Relations. In: *Proceedings of ACL workshop on the linguistic annotation*, pages 132-139, Prague, Czech Republic.

Jijkoun, V., Rijke D. M. 2004. Enriching the Output of a Parser Using Memory-based Learning. In: *Proceedings of the 42nd ACL*, pages 311-318, Barcelona, Spain.

Kiss, T., Strunk, J. 2006. Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics*, 32(4):485-525.

Kudo, T., Matsumoto, Y. 2001. Chunking with Support Vector Machines. In: *Proceedings of the NAACL 2001*, pages 1-8, Pittsburgh, USA.

Nocedal, J., Wright, S. J. 1999. Numerical Optimization. Springer.

Lafferty, J., McCallum, A., Pereira, F. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proceedings of ICML 2001*, pages 282-289, Williamstown, USA.

Levin, B. 1993. *English Verb Classes and Alternations: A preliminary Investigation.* The University of Chicago Press, USA.

Marcus, M., Kim, G., Marcinkiewicz, A. M., Macintyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B. 1994. The Penn Treebank: Annotating Predicate Argument Structure. In: *Proceedings of ARPA Human Language Technology Workshop*, San Francisco, USA.

Mccallum, A., Freitag, D., Pereira, F. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In: *Proceedings of ICML 2000*, pages 591-598, Stanford University, USA.

Merlo, P., Ferrer, E. E. 2006. The Notion of Argument in Prepositional Phrase Attachment. *Computational Linguistics*, 32(3):341-378.

Merlo, P., Musillo, G. 2005. Accurate Function Parsing. In: *Proceedings of EMNLP 2005*, pages 620-627, Vancouver, Canada.

Qin, Y., Yuan, C., Sun, J., Wang, X. 2008. BUPT Systems in the SIGHAN Bakeoff 2007. In: *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*, pages 94-97, Hyderabad, India.

Rabiner, L. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: *Proceedings of the IEEE*, 77(2):257-286.

Ramshaw, L., Marcus, M. 1995. Text Chunking Using Transformation Based Learning. In: *Proceedings of ACL Third Workshop on Very Large Corpora*, pages 82-94, Cambridge MA, USA.

Swier, R., Stevenson, S. 2004. Unsupervised Semantic Role Labelling. In: *Proceedings of EMNLP-2004*, pages 95-102, Barcelona, Spain.

Tsochantaridis, T., Hofmann, T., Joachims, T., Altun, Y. 2004. Support Vector Machine Learning for Interdependent and Structured Output Spaces. In: *Proceedings of ICML 2004*, pages 823-830, Banff, Canada.

Wang, M., Sagae, K., Mitamura, T. 2006. A Fast, Accurate Deterministic Parser for Chinese. In: *Proceedings of the 44th ACL*, pages 425-432, Sydney, Australia.

Xue, N., Xia, F., Huang, S., Kroch, T. 2000. The Bracketing Guidelines for the Chinese Treebank. *IRCS Tech., rep., University of Pennsylvania.*

Zhao, Y., Zhou, Q. 2006. A SVM-based Model for Chinese Functional Chunk Parsing. In: *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 94-10, Sydney, Australia1.

Zhou, Q., Zhan, W., Ren, H. 2001. Building a Large-scale Chinese Chunkbank (in Chinese). In: *Proceedings of the 6th Joint Conference of Computational Linguistics of China*, Taiyuan, China.