Multi-view Chinese Treebanking

Likun Qiu[1,2,3]**, Yue Zhang**[1]**, Peng Jin**[4] **and Houfeng Wang**[2]
[1]Singapore University of Technology and Design, Singapore
[2]Institute of Computational Linguistics, Peking University, China
[3]School of Chinese Language and Literature, Ludong University, China
[4]Lab of Intelligent Information Processing and Application, Leshan Normal University, China
{qiulikun,jandp,wanghf}@pku.edu.cn, yue_zhang@sutd.edu.sg

Abstract

We present a multi-view annotation framework for Chinese treebanking, which uses dependency structures as the base view and supports conversion into phrase structures with minimal loss of information. A multi-view Chinese treebank was built under the proposed framework, and the first release (PMT 1.0) containing 14,463 sentences is be made freely available. To verify the effectiveness of the multi-view framework, we implemented an arc-standard transition-based dependency parser and added phrase structure features produced by the phrase structure view. Experimental results show the effectiveness of additional features for dependency parsing. Further, experiments on dependency-to-string machine translation show that our treebank and parser could achieve similar results compared to the Stanford Parser trained on CTB 7.0.

1 Introduction

Phrase structures (PS) and dependency structures (DS) are two of the most popular grammar formalisms for statistical parsing (Collins, 2003; Charniak, 2000; McDonald et al., 2005; Nivre, 2006; Petrov and Klein, 2007; Zhang and Clark, 2008). While DS trees emphasize the grammatical relation between heads and dependents, PS trees stress the hierarchical constituent structures of sentences. Several researchers have explored DS and PS simultaneously to enhance the quality of syntactic parsing (Wang and Zong, 2010; Farkas and Bohnet, 2012; Sun and Wan, 2013) and tree-to-string machine translation (Meng et al., 2013), showing that the two types of information complement each other for NLP tasks.

Most existing Chinese and English treebanks fall into the phrase structure category, and much work has been done to convert PS into DS (Magerman, 1994; Collins et al., 1999; Collins, 2003; Sun and Jurafsky, 2004; Johansson and Nugues, 2007; Duan et al., 2007; Zhang and Clark, 2008). Research on statistical dependency parsing has frequently used dependency treebanks converted from phrase structure treebanks, such as the Penn Treebank (PTB) (Marcus et al., 1993) and Penn Chinese Treebank (CTB) (Xue et al., 2000). However, previous research shows that dependency categories in converted treebanks are simplified (Johansson and Nugues, 2007), and the widely used head-table PS to DS conversion approach encounters ambiguities and uncertainty, especially for complex coordination structures (Xue, 2007). The main reason is that the PS treebanks were designed without consideration of DS conversion, leading to inherent ambiguities in the mapping, and loss of information in the resulting DS treebanks. To minimize information loss during treebank conversions, a treebank could be designed by considering PS and DS information simultaneously; such treebanks have been proposed as *multi-view* treebanks (Xia et al., 2009). We develop a multi-view treebank for Chinese, which treats PS and DS as different views of the same internal structures of a sentence.

We choose the DS view as the base view, from which PS would be derived. Our choice is based on the effectiveness of information transfer rather than convenience of annotation (Rambow, 2010; Bhatt and Xia, 2012). Research on Chinese syntax (Zhu, 1982; Chen, 1999; Chen, 2009) shows that the phrasal category of a constituent can be derived from the phrasal categories of its immediate subconstituents and

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,
pages 257–268, Dublin, Ireland, August 23-29 2014.

PKU POS	Our POS
Ag, a, ad, ia, ja, la	a (adjective)
Bg,b, ib, jb, jm, lb	b (distinguishing words)
Dg, d, dc, df, id, jd, ld	d (adverb)
m, mq	m(number)
n, an, in, jn, ln, Ng, vn, nr, kn	n (noun)
Qg,q, qb, qc, qd, qe, qj, ql, qr, qt, qv, qz	q (measure word)
Rg,r, rr, ry, ryw, rz, rzw	r (pronoun)
Tg, t, tt	t (temporal noun)
u, ud, ue, ui, ul, uo, us, uz, Ug	u (auxiliary word)
v, iv, im, jv, lv, Vg, vd, vi, vl, vq,vu, vx, vt,kv	v (verb)
w, wd, wf, wj, wk, wky, wkz, wm,wp, ws, wt, wu, ww, wy, wyy, wyz	w (punctuation)

Table 1: Mapping from PKU POS to our POS.

the dependency categories between them (for terminal words, parts-of-speech can be used as phrasal categories). Consequently, in Chinese, the canonical PS, containing information of constituent hierarchies and phrasal categories, can be derived naturally from the canonical DS. As Xia et al. (2009) stated, a rich set of dependency categories should be designed to ensure lossless conversion from DS to PS. When the information of PS has been represented in DS explicitly or implicitly, we can convert DS to PS without ambiguity (Rambow et al., 2002).

Given our framework, a multi-view Chinese treebank, containing 14,463 sentences and 336K words, is constructed. This main corpus is based on the Peking University People's Daily Corpus. We name our treebank the Peking University Multi-view Chinese Treebank (PMT) release 1.0. To verify the usefulness of the treebank for statistical NLP, a transition-based dependency parser is implemented to include PS features produced in the derivation process of phrasal categories. We perform a set of empirical evaluations, with experimental results on both dependency parsing and dependency-to-string machine translation showing the effectiveness of the proposed annotation framework and treebank. We make the treebank, the DS to PS conversion script and the parser freely available.

2 Annotation Framework

2.1 Part-of-speech Tagset

Our part-of-speech (POS) tagset is based on the Peking University (PKU) People's Daily corpus, which consists of over 100 tags (Yu et al., 2003). We simplify the PKU tagset by syntactic distribution. The simplified tagset contains 33 POS tags. The mapping from the original PKU POS to our simplified POS is shown in Table 1. For instance, *Ag* (adjective morpheme), *ad* (adjective acting as an adverb), *ia* (adjective idioms), *ja* (adjective abbreviation) and *la* (temporary phrase acting as an adjective) are all mapped to one tag *a* (adjective). A set of basic PKU POS tags, including *c* (conjunction), *e* (interjection), *f* (localizer), *g* (morpheme), *h* (prefix), *i* (idiom), *j* (abbreviation), *k* (suffix), *l* (temporary phrase), *nr* (personal name), *nrf* (family name), *nrg* (surname), *ns* (toponym), *nt* (organization name), *nx* (non-Chinese noun), *nz* (other proper noun), *o* (onomonopeia), *p* (preposition), *q* (measure word), *r* (pronoun), *s* (locative), *x* (other non-Chinese word), *y* (sentence final particle), *z* (state adjective), are left unchanged.

2.2 Dependency Category Tagset

In a DS, the modifier is tagged with a dependency category, which denotes the role the modifier plays with regard to its head. The root word of a sentence is dependent on a virtual root node *R* and tagged with the dependency category *HED*. Table 2 lists the 32 dependency categories used in our annotation guideline. These categories are designed in consideration of PS conversion with minimal ambiguities, and can be classified according to the following criteria:

(1) whether the head dominates a compound clause (i.e. has an IC modifier) in the PS view. According to this, dependency categories can be *cross-clause* or *in-clause*. For instance, in Figure 1, the last punctuation (。) is labeled with the *cross-clause* tag *PUS*, and its head dominates an *IC* modifier. (2) the relative position of the modifier to the head. According to this, dependency categories can be *left*, *right* or *free*. For instance, the *LAD, SBV, ADV, COS, DE and ATT* labels in Figure 1 are all *left*. The *VOB* label

Tag	Description	Tag	Description
ACT	action object	LAD	left additive
ADV	adverbial	MT	modality and time
APP	appositive element	NUM	number
ATT	attribute	POB	propositional object
CMP	complement	PUN	punctuation
COO	other coordination element	PUS	cross-clause punctuation
COS	share-right-child coordination element	QUC	post-positional quantity
DE	de (modifier of 的(special function word))	QUCC	non-shared post-positional quantity
DEI	dei (modifier of 得(special function word))	QUN	quantity
DI	di (modifier of 地(special function word))	RAD	right additive
FOC	focus	RADC	non-shared right additive
HED	root of a sentence	RED	reduplicate element
IC	independent clause	SBV	subject
IOB	indirect object	TPC	topic
IS	independent structure	VOB	direct object
ISC	non-shared independent structure	VV	serial verb construction

Table 2: Proposed dependency category set.

is *right*, while the *PUS, PUN, IC* labels are *free* and can lie on both sides. (3) whether the modifiers of a head follows the right-to-left order when combined with the head for deriving the PS structure. According to this, dependency categories can be *special* (not following the right-to-left order) or *common*. For instance, in Figure 1, the word "观察 (observe)" was labeled with the *special left* tag *COS*, because it is combined with its head "体贴 (consider)" before "体贴 (consider)"'s VOB modifier on the right.

Combining the three perspectives, the 32 dependency categories can be classified into 8 classes. Categories in different classes have different priorities when attached to the head word during PS conversion.

(1) *Special left* (2 labels): COS and RED. If there is a word tagged with the special left category, all the words between this word and its head word should be taken as *special left*.

(2) *Common left* (13 labels): ADV, APP, ATT, DE, DI, FOC, NUM, QUN, SBV, TPC, VV, PUN and IS. For instance, "就要 (must)" in Figure 1 is labeled with the *common left* tag ADV and follows the right-to-left order, being combined with its head "善于 (be good at)" after "体贴 (consider)".

(3) *Common left cross-clause* (5 labels): ADV, SBV, LAD, TPC and IS. A *common left cross-clause* modifier can also act like common left in-clause, but not vice versa.

(4) *Common right* (7 labels): ACT, CMP, DEI, IOB, MT, POB and VOB. For instance, the word "心理 (psychology)" in Figure 1 is labeled with *VOB* and follows the right-to-left order.

(5) *Special right* (4 labels): QUC, RAD, PUN, IS. In particular, PUN and IS are common categories when appearing on the left side but special categories on the right side of the head.

(6) *Special right* (attached before COO) (3 labels): QUCC, RADC and ISC. These categories differ from those in the previous class in that they would be combined to the head before COO modifiers.

(7) *Free cross-clause* (2 labels): IC, PUS. IC is a clausal category and so can be used to connect two clauses. PUS denotes cross-clause punctuations.

(8) *Common left coordination* (2 labels): COO and LAD.

2.3 Rules for Annotating Punctuations

To resolve the ambiguity of finding the head of a punctuation, we make the following rules.

(1) Coupled punctuations (e.g. brackets and quotation marks) take the head word of the phrase between the two punctuations as their head.

(2) Full stops, question marks, exclamatory marks and semicolons take the topmost head word (without violating projectivity) on their left as their heads.

(3) Commas take the nearest word on the right with HED or IC, or the topmost head words on the right (if there is no right node tagged with HED or IC), or the nearest words on their left tagged with HED or IC as their heads, all under the condition of not breaking projectivity.

(4) Colons take the topmost head word (without violating projectivity) on their right as their heads.

(5) Slight-pause marks (、) take the head of the COO or COS constituent on their left as their heads.

3 Automatic Derivation of Phrase Function and Hierarchy

In our treebank, DS is represented explicitly and PS implicitly. The conversion from DS to PS consists of two steps. First, a binary PS hierarchy is generated bottom-up according to the DS. Second, each non-terminal node in the hierarchy is tagged with a phrasal tag (e.g. NP, VP) based on manual rules. We adopt the PS tagset of the CTB (Xue et al., 2000) for our treebank.

3.1 Derivation of Phrase Hierarchy

3.1.1 Derivation Algorithm

The PS trees in our grammar are binary-branching, making the derivation of hierarchical PS from DS relatively straightforward. With leaf nodes being pre-terminals, a PS is derived bottom-up by recursive combinations of neighbouring spans according to the dependency links in a sentence. In this process, a head word is always combined with the nearest modifier that is currently not in the constituents it dominates. The only ambiguities lie in the orders in which neighbouring PS are combined to form a larger PS, which can be denoted as (A (B C)) versus ((A B) C), with A, B, and C being three neighbouring spans. For the above ambiguity to exist, the head word for each span must bare the dependency links ($A \curvearrowleft B \curvearrowright C$), with the head word of B being the head of those of A and C.

In most cases, the (A (B C)) structure is chosen. An intuitive example is that a verb is first combined with the object (VOB, a *common right* category) to form a VP, before being combined with the subject (SBV, a *common left* category) to form an IP. One example of ((A B) C) structures is the coordination structure shown in Figure 1, where the spans headed by “观察 (observe)” and “心理 (psychology)” are combined after those by “观察 (observe)” and “体贴 (consider)”, due to the fact that “心理 (psychology)” is a shared object to the coordinated verbs, linked by a COS (a *special left* category) arc. In general, the modifiers of a given head are attached according the following priorities:

(1) the *special left* category $>$ (2) the *common right* category $>$ (3) the *common left* category $>$ (4) the *special right* category before COO $>$ (5) the *common left* coordination category $>$ (6) the other *special right* category $>$ (7) the *free cross-clause* clausal category (IC) $>$ (8) the *common left cross-clause* category $>$ (9) the *free cross-clause* punctuations (PUS).

3.1.2 A Case Study: Generating the Hierarchy of Coordination Structure

We take coordination structures as an example to illustrate the PS hierarchy generation process. Typically, researchers treat the rightmost conjunct as the head of a coordinate structure. However, doing so introduces modifier scope ambiguities when modifiers are also attached to the rightmost head. Vice versa, treating the leftmost conjunct as the head will lead to ambiguities when modifiers attached to the left head (Che et al., 2012). Another choice is treating the conjunction as the head (Huang et al., 2000; Xue, 2007). However, this is usually not preferred since it makes parsing more difficult and a choice still has to be made between the left and right elements when there is no conjunction in a coordinate structure (Xue, 2007). Our strategy is as follows: (1) Choose the rightmost conjunct as the head to eliminate the ambiguities when the modifiers are attached to the left; (2) Classify coordinate structures into common coordinate structures (COO) and sharing-right-child coordinate structures (COS). COO words are taken as *common left* nodes (as shown in Figure 2), while COS words are *special left* nodes (as shown in Figure 1). Doing so avoids the aforementioned scope ambiguities for modifiers.

3.2 Derivation of Phrasal Category

Several Chinese linguists discuss the issue of deriving phrasal categories from the syntactic categories of the PS and DS context. Both Zhu (1982) and Chen (1999) state that if two phrases have constituents with the same phrasal categories and the dependency types between them are also the same, the phrasal categories of their combinations must be the same. Consequently, it is natural to derive the category of a phrase from the phrasal categories of the immediate constituents and the dependency type between the constituents. We make a set of rules for the derivation, each being a DS pattern/phrasal type pair. The DS pattern is a modifier-head link with associated information such as the dependency category (DepCate)

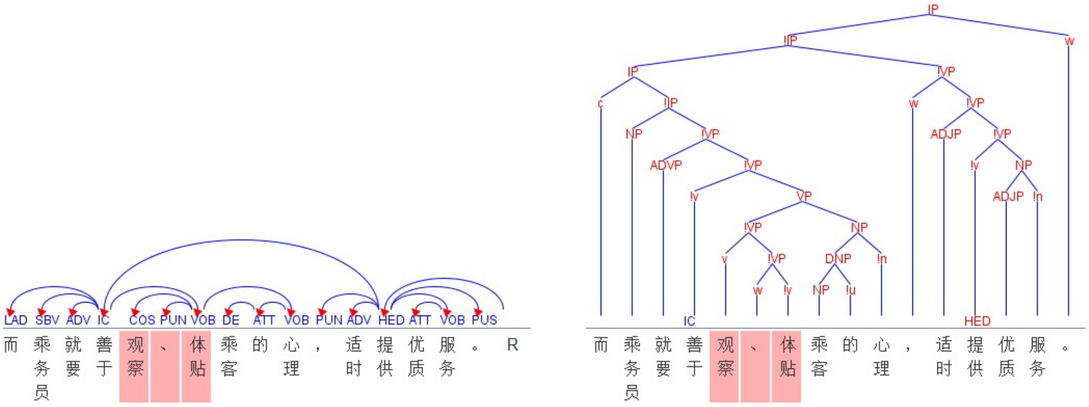

LAD SBV ADV IC COS PUN VOB DE ATT VOB PUN ADV HED ATT VOB PUS
而 乘务员 就要 善于 观察 、 体贴 乘客 的 心理 ， 适时 提供 优质 服务 。 R
IP
!IP
w
IP
!VP
c
!IP
w
!VP
NP
!VP
ADJP
!VP
ADVP
!VP
!v
NP
!v
VP
ADJP
!n
!VP
NP
v
!VP
DNP
!n
w
!v
NP
!u
IC
HED
而 乘务员 就要 善于 观察 、 体贴 乘客 的 心理 ， 适时 提供 优质 服务 。

Figure 1: An instance of DS-PS conversion (而 (moreover) 乘务员 (crew) 就要 (must) 善于 (be good at) 观察 (observe) 体贴 (consider) 乘客 (passenger) 的 ('s) 心理 (psychology)， 适时 (timely) 提供 (provide) 优质 (quality) 服务 (service)). "!" denotes the head constituent.

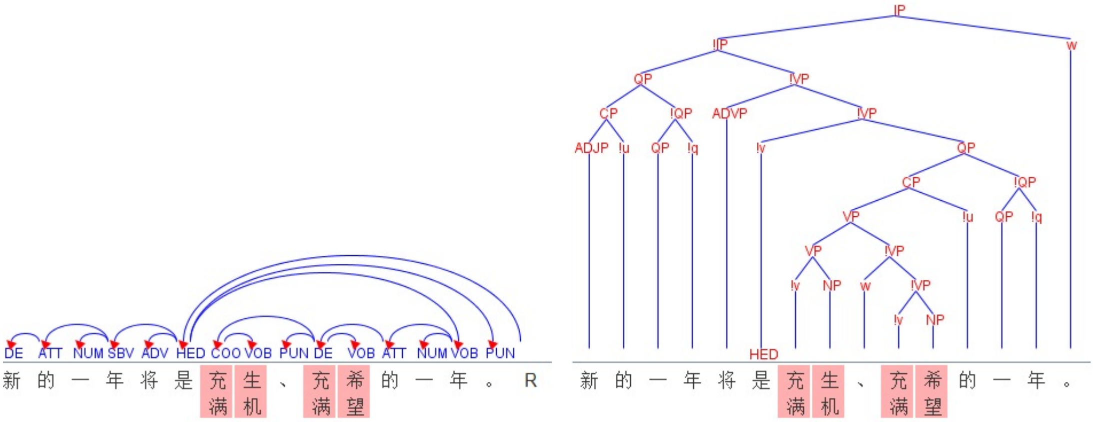

DE ATT NUM SBV ADV HED COO VOB PUN DE VOB ATT NUM VOB PUN
新 的 一 年 将 是 充满 生机 、 充满 希望 的 一 年 。 R
IP
!IP
w
QP
!VP
CP
!QP
ADVP
!VP
ADJP
!u
QP
!q
!v
QP
CP
!QP
VP
!u
QP
!q
VP
!VP
!v
NP
w
!VP
!v
NP
HED
新 的 一 年 将 是 充满 生机 、 充满 希望 的 一 年 。

Figure 2: A second instance of DS-PS conversion (新 (new) 的 (de, an auxiliary word) 一 (one) 年 (year) 将 (will) 是 (be) 充满 (be full of) 生机 (vitality)、 充满 (be full of) 希望 (hope) 的 (de, an auxiliary word) 一 (one) 年 (year)). "!" denotes the head constituent.

and the phrasal categories (POS tags for terminial nodes) of the subphrases that the modifier and head dominates. Some high-frequency rules are listed in Table 3.

For instance, the phrasal category of 充满 (be full of) 生机 (vitality) in Figure 2 is *VP* using the rule (v-NP-VOB, VP). Executing the derivation algorithms in Section 3.1 and derivation rules in Section 3.2, a DS in the proposed framework can be converted into corresponding PS, as shown in Figure 1 and 2.

4 The Annotation Process of PMT

According to the proposed schema, we constructed the multi-view Chinese treebank (PMT), version 1.0, which contains about 14,463 sentences and 336K words, and supports both the PS view and DS view. Our treebanking is based on the work of Yu et al. (2003), who built a segmented and POS-tagged Chinese corpus (the PFR Corpus), and released a sub-corpus containing about 1.1M words for free[1]. We choose the previous 14,463 sentences from the corpus, follow the original word segmentation standard but simplify the POS tagset according to the mapping rules described in Section 2.1. Then each sentence is annotated into a projective dependency tree according to the annotation framework described in this paper.

To speed up the annotation, a statistical dependency parser is used to give automatic parse trees and annotators are required to check each tree on a visualized annotation platform, which supports detecting

[1] http://klcl.pku.edu.cn/ShowNews.aspx?id=110

HCate	MCate	DepCate	PCate	HCate	MCate	DepCate	PCate
v	NP	VOB	VP	VP	NP	SBV	IP
IP	w	PUS	IP	NP	w	PUN	NP
n	NP	ATT	NP	VP	IP	IC	IP
p	NP	POB	PP	n	CP	ATT	NP
n	DNP	ATT	NP	NP	CP	ATT	NP
u	VP	DE	CP	NP	DNP	ATT	NP
NP	n	COO	NP	v	IP	VOB	VP
VP	n	SBV	IP	VP	v	ADV	VP
u	IP	DE	CP	NP	NP	ATT	NP
p	LCP	POB	PP	VP	c	LAD	VP
VP	d	ADV	VP	VP	PP	ADV	VP
IP	IP	IC	IP	VP	NP	VOB	VP
u	NP	DE	DNP	VP	r	SBV	IP
NP	NP	COO	NP	VP	r	SBV	IP
NP	n	ATT	NP	VP	VP	IC	IP

Table 3: Some rules for generating phrasal categories. HCate, MCate and PCate denote the phrasal category of the head subphrase, the modifier subphrase and the combined phrase, respectively.

invalid derivation from DS to PS.

For quality control, a detailed annotation guideline is provided with abundant instances for different types of syntactic structures in Mandarin Chinese. More information of the guideline can be found in an extended version of this paper. In addition, we adopt the annotation strategy for the construction of the Penn Chinese Treebank (Xue et al., 2000) — one annotator examines an automatic parse tree first, and a second annotator verifies the annotation of the first annotator.

5 A Transition-Based Parser for Multi-view Treebank

In order to demonstrate the usefulness of our treebank in comparison with existing Chinese treebanks, we perform empirical analysis to the treebank, by the statistical dependency parsing and dependency-to-string machine translation tasks. Several researchers explored joint DS and PS information to enhance the quality of syntactic parsing (Wang and Zong, 2010; Farkas and Bohnet, 2012; Sun and Wan, 2013). Most tried to combine the outputs of constituent and dependency parsers by stacking or bagging. Since our treebank is multi-view, it is possible to combine DS features and PS features directly in the decoding process.

We implemented an arc-standard transition-based dependency parser (Nivre, 2008) based on the arc-eager parser of Zhang and Nivre (2011), which is a state-of-the-art transition-based dependency parser (Zhang and Nivre, 2012). It is more reasonable to derive the phrasal category of a phrase after the complete subtree (phrase) rather than partial subtree headed by a word has been built. The arc-standard parser differs from the arc-eager parser in that it postpones the attachment of right-modifiers until the complete subtrees headed by the modifiers themselves have been built. Because of this, we add PS features into an arc-standard parser rather than an arc-eager one.

The parser processes a sentence from left to right, using a stack to maintain partially built derivations and a queue to hold next incoming words. Three transition actions (LEFT, RIGHT and SHIFT) are defined to consume input words from the queue and construct arcs using the stack (Nivre, 2008):

LEFT pops the second top item off the stack, and adds it as a modifier to the top of the stack;

RIGHT pops the top item off the stack, and adds it as a modifier to the second top of the stack;

SHIFT removes the front of the queue and pushes it onto the top of the stack.

Table 4 show the feature templates of our parser, most of which are based on those of Zhang and Nivre (2011). The contextual information consists of the top four nodes of the stack (S_3, S_2, S_1 and S_0), the next three input words (N_0, N_1 and N_2), the left and right children (ld, rd) of these nodes, and the distance between S_0 and S_1. Word and POS information from the context are manually combined.

Due to the multi-view nature of our treebank, the DS parser can be extended naturally to incorporate PS information. Further, because our PS is binary branching, each constituent corresponds to a dependency link. In the decoding process, we derive the phrasal category c of a subtree whenever a dependency link

features of stack top	$S_0wt; S_0w; S_0t; S_1wt; S_1w; S_1t; S_2wt; S_2w; S_2t; S_3wt; S_3w; S_3t; N_0wt;$
features of next input	$N_0w; N_0t; N_1wt; N_1w; N_1t; N_2wt; N_2w; N_2t;$
bigram features	$S_0wS_1w; S_0wS_1t; S_0tS_1w; S_0tS_1t; S_0wN_0w; S_0wN_0t; S_0tN_0w; S_0tN_0t;$
children features of S_0	$S_0ldw; S_0ldt; S_0ldwt; S_0ldd; S_0rdw; S_0rdt; S_0rdwt; S_0rdd;$
children features of S_1	$S_1ldw; S_1ldt; S_1ldwt; S_1ldd; S_1rdw; S_1rdt; S_1rdwt; S_1rdd;$
distance features	$S_0wDistance(S_0, S_1); S_0tDistance(S_0, S_1); S_1wDistance(S_0, S_1); S_1tDistance(S_0, S_1);$
PS features	$S_0c; S_1c; S_0cS_1c; S_0wS_1c; S_0tS_1c; S_0wS_1dS_1c; S_1wS_0c; S_1tS_0c; S_1wS_0dS_0c; S_0cS_1cS_0S_1c$

Table 4: Transition-based feature templates for the arc-standard dependency parser. *w*=word; *t*=POS tag. *d*=dependency category. *c*=phrasal category.

is established, using the derivation rules in Table 3. Using *c* and its combination with other features, we can produce several PS features, as shown in Table 4. By this simple extension of features, we arrive at an efficient linear-time joint DS and PS parser.

6 Experiments

6.1 Syntactic Parsing

PMT 1.0 contains all the articles of People's Daily from January 1st to January 10th, 1998. Sentences 12001-13000 and 13001-14463 are used as the development and test set, respectively. The remaining sentences are used as training data.

Several state-of-the-art statistical parsers, including Mate-tools (Bohnet, 2010)[2], BerkeleyParser (Petrov and Klein, 2007)[3], ZPar-dep (Zhang and Nivre, 2011) and ZPar-con (Zhang and Clark, 2009; Zhu et al., 2013)[4] are used for comparison. We used the gold segmentation, and the Stanford POS tagger (Toutanova et al., 2003) (version 3.3.1) to provide automatic POS tags for all the experiments. The POS tagger was trained on the PKU corpus (Yu et al., 2003) containing articles of People's Daily from January 2000 to June 2000. It achieved a 95.78% precision on the PMT. In the baseline parser (Ours-standard), the feature templates in Table 4 except the PS features are used. We refer to the parser after adding PS features as Ours-PS. The results of dependency (ZPar-eager, Ours-standard, Ours-PS and Mate-tools) and constituent parsers (BerkeleyParser and ZPar-con) are measured by the unlabeled accuracy score (UAS), labeled accuracy score (LAS) and bracketing f-measure (BF), respectively.

We display the parsing results in Table 5. Our dependency parser (Ours-PS) outperforms the baseline parser (Ours-standard) with a 0.47% increase in UAS. For additional evaluation, we also converted the DS trees parsed by the dependency parsers to PS using the conversion procedure in Section 3, in order to compare the results of dependency parsers and constituent parsers. The three ZPar-based dependency parsers gave higher accuracies than the two state-of-the-art constituent parsers. In particular, the DS2PS outputs of Ours-PS parser outperforms the PS outputs of Berkeley Parer with 0.62% higher BF.

Both Zhang and Clark (2011) and Petrov and McDonald (2012) show that DS trees converted from the outputs of PS parsers outperform those produced directly by DS parsers trained on DS conversions of the CTB. Interestingly, our evaluation on the PMT gave results in the opposite direction: parsers trained on the DS treebank outperforms parsers trained on the PS conversion. One possible reason is that parser errors can be hidden in the conversion process. Take the sentence in Figure 3 for example. Figure 3(a) shows the correct PS while Figure 3(b) shows an incorrect parser output. In particular, "黎明 (dawn)" is put under the incorrect constituent. When converted into DS, both lead to the correct link, with "黎明 (dawn)" being the SBV modifier of "降临 (come)" (Figure 3(c)). As a result, the PS parser error is erased in the conversion into DS. The same can happen in DS to PS conversion.

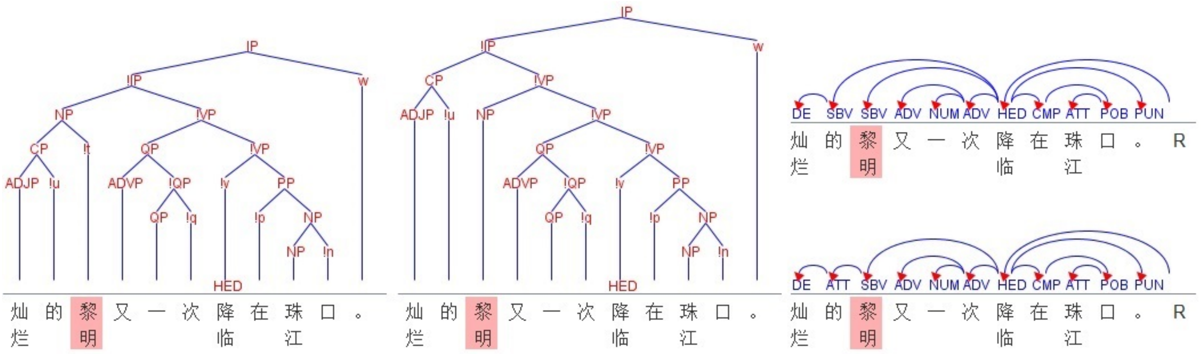
6.2 Dependency-to-string Machine Translation

We compare the effects of our treebank and the Stanford dependencies converted from CTB on machine translation, using the dependency-to-string system of Xie et al. (2011). Our training corpus consists of

[2] https://code.google.com/p/mate-tools/

[3] http://code.google.com/p/berkeley-parser-analyser/

[4] http://sourceforge.net/projects/zpar/


IP
IP
w
NP
IVP
CP
!t
QP
IVP
ADJP
!u
ADVP
!QP
!v
PP
QP
!q
!p
NP
NP
!n
HED
灿烂 的 黎明 又 一 次 降临 在 珠江 口 。
CP
ADJP
!u
NP
DE SBV SBV ADV NUM ADV HED CMP ATT POB PUN
灿烂 的 黎明 又 一 次 降临 在 珠江 口 。 R
DE ATT SBV ADV NUM ADV HED CMP ATT POB PUN

(a) Correct PS: "灿烂 (bright) 的 (de, auxiliary word) 黎明 (dawn) is a subphrase.

(b) Incorrect PS: "灿烂 的 " and "黎明 " are not in the same subphrase.

(c) Incorrect DS (above) and correct DS (below): 黎明 (dawn) is the SBV modifier of 降临 (come).

Figure 3: An instance where PS parser error is erased in the PS to DS conversion (灿烂 (bright) 的 (de, an auxiliary word) 黎明 (dawn) 又 (again) 一 (one) 次(time) 降临 (come) 在 (in) 珠江 (the Pearl River) 口 (estuary)). “!” denotes the head constituent.

Dependency Parsing			Constituent Parsing		Constituent Parsing(DS2PS)	
Parsers	UAS	LAS	len<=40 words	Unlimited	len<=40 words	Unlimited
Mate-tools	82.98	79.37	/	/	84.77	83.43
ZPar-dep	82.73	80.20	/	/	85.47	84.33
Ours-standard	82.81	80.04	/	/	85.53	84.47
Ours-PS	**83.28**	**80.50**	/	/	**85.92**	**84.84**
Berkeley Parser	/	/	85.25	84.22	/	/
ZPar-con	/	/	85.02	84.12	/	/

Table 5: Parsing results on our treebank using automatic POS-tags.

31K Chinese-English sentence pairs from the Xinhua Corpus (Liu et al., 2006), and we used NIST MT Evaluation 2006 test set as the development set, and the NIST 2003 (MT03), 2004 (MT04) and 2005 (MT05) test sets as the test sets. For Stanford dependency trees, we parsed the source sentences with the Stanford Parser (Chang et al., 2009) (version 3.3.1), which was trained on CTB 7.0. For the PMT treebank, we used the Ours-PS parser, trained with 14000 sentences (the last 463 sentences are used as development data for the parser). All the MT configurations are the same as Xie et al. (2011).

The results are shown in Table 6. The Chinese-English translation outputs using our parser and treebank are slightly lower but comparable to those using the Stanford Parser. Note that our treebank contains 336K words on People’s Daily, while the CTB 7.0 contains about 1.19M words, most on Xinhua, the source of the MT training and test data. This result to some degree demonstrates the usefulness of our treebank for NLP applications, in comparison with a well-established treebank.

7 Related Work

PS Treebanks and DS Conversion PTB (Marcus et al., 1993) and CTB (Xue et al., 2000) are the most widely used treebanks for English and Chinese in the literature. Both are in PS. For conversion from PS to DS, a head-table approach (Magerman, 1994; Collins, 2003; Yamada and Matsumoto, 2003; Sun and Jurafsky, 2004; Nivre, 2006; Johansson and Nugues, 2007; Duan et al., 2007; Zhang and Clark, 2008) is widely used. However, the reliability of head tables has been questioned (Xue, 2007). Xue (2007) proposed a novel approach that better exploits the structural information in the CTB and pointed out that the results of the approach and the widely used Penn2Malt tools[5] agree only 60.6% in terms of unlabeled dependency. The coordination structures, in particular, are not properly converted by Penn2Malt.

DS Treebanks and PS Conversion An existing DS treebank for Chinese is the Chinese Dependency Treebank (Che et al., 2012), which is not designed as a multi-view treebank. For conversion from DS to PS, Xia and Palmer (2001) compare three algorithms. These algorithms do not use a rich set of

[5] http://stp.lingfil.uu.se/ nivre/research/Penn2Malt.html

Parsers	Treebank	MT03(BLEU4)	MT04(BLEU4)	MT05(BLEU4)
Stanford Parser	CTB 7.0	28.23	29.00	25.72
Ours-PS	PMT 1.0	27.73	28.71	25.20

Table 6: Results of dependency-to-string machine translation.

dependency categories, only distinguishing arguments and modifiers. Xia et al. (2009) propose a DS-to-PS algorithm, which assumes that a given DS is identical to a flattened version of the desired PS, and then introduce a set of conversion rules. Their error analysis show that coordination and punctuation amount to about 32.1% of conversion errors, while other errors fall into missing content in DS and inconsistency in the target treebank (PTB). This analysis demonstrates that coordination and punctuation should be tackled carefully for the conversion between PS and DS, which we do in the design of our treebank. Bhatt et al. (2011) presented three scenarios arising in the conversion of DS into PS. Bhatt and Xia (2012) further described 7 phenomena of incompatibility in the conversion from DS to PS, mainly involving the annotation of empty categories, yet coordination structure and punctuation were not discussed.

Multi-view Treebanks The Tiger (Brants et al., 2002) and TüBa-D/Z (Telljohann et al., 2003) treebanks for German seek to explicitly represent both PS and DS by labeling both nodes and edges in the syntactic tree. For these treebanks, both dependency categories and phrasal categories have been annotated explicitly. The English side of the Czech-English parallel corpus is annotated and linked also as both PS (original PTB annotation) and DS (Hajic et al., 2012), while the DS is a conversion of the original PS. Our multi-view treebank is different in that dependency categories and phrasal categories derive from each other. The Hindi/Urdu treebank (Xia et al., 2009; Palmer et al., 2009; Bhatt et al., 2009) can be taken as a multi-view treebank. Its PS view is derived automatically from the DS. However, the converted PS is not a PS with a full hierarchy but a flattened one (Xia et al., 2009).

8 Conclusion

We presented an DS-based multi-view annotation framework, and built a Chinese treebank according to the framework and an arc-standard transition-based dependency parser that exploits the multi-view nature of the treebank. We used SMT as an example to demonstrate the usefulness of our treebank for NLP applications. Experiments showed that the proposed treebank and parser can give similar results to the Stanford Parser trained on CTB 7.0. We make our treebank (PMT 1.0) (`http://klcl.pku.edu.cn/ResourceList.aspx`), the DS to PS conversion script and the proposed parser (`http://sourceforge.net/projects/zpar/`) freely available.

Acknowledgments

We gratefully acknowledge the invaluable assistance of Jun Xie in helping us make experiments on dependency-to-string machine translation. We thank Ji Ma for helpful suggestions and comments, and Yijia Liu for providing a generic arc-standard dependency parser. We also thank all the students from Ludong University and Leshan Normal University for annotating the large-scale treebank. Special thanks to Xiaoyan Wang, Na Wei and Huarong Ni from Ludong University. Finally, we thank the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the Singapore Ministry of Education (MOE) AcRF Tier 2 grant T2MOE201301, the National Natural Science Foundation of China (No. 61103089, No. 61373056), National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101), Major National Social Science Fund of China (No. 12&ZD227), Scientific Research Foundation of Shandong Province Outstanding Young Scientist Award (No. BS2013DX020) and Humanities and Social Science Projects of Ludong University (No. WY2013003).

References

Rajesh Bhatt and Fei Xia. 2012. Challenges in converting between treebanks: a case study from the hutb. In *Proceedings of META-RESEARCH Workshop on Advanced Treebanking, in conjunction with LREC-2012, Istanbul, Turkey*.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189.

Rajesh Bhatt, Owen Rambow, and Fei Xia. 2011. Linguistic phenomena, analyses, and representations: Understanding conversion between treebanks. In *Proceedings of IJCNLP 2011*, pages 1234–1242.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING 2010*, pages 89–97.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, volume 168.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D Manning. 2009. Discriminative reordering with chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the NAACL 2000*, pages 132–139.

Wanxiang Che, Li Zhenghua, and Liu Ting. 2012. *Chinese Dependency Treebank 1.0*. Linguistic Data Consortium.

Baoya Chen. 1999. *Chinese Linguistic Methodology in the 21th Century, 1898-1998*. Shandong Education Publishing House.

Baoya Chen. 2009. *Contemporary Linguistics*. Higher Education Press.

Michael Collins, Lance Ramshaw, Jan Hajič, and Christoph Tillmann. 1999. A statistical parser for czech. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 505–512. Association for Computational Linguistics.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.

Xiangyu Duan, Jun Zhao, and Bo Xu. 2007. Probabilistic parsing action models for multi-lingual dependency parsing. In *Proceedings of EMNLP-CoNLL 2007*, pages 940–946.

Richárd Farkas and Bernd Bohnet. 2012. Stacking of dependency and phrase structure parsers. In *Proceedings of COLING 2012*, pages 849–866.

Jan Hajic, Eva Hajicová, Jarmila Panevová, Petr Sgall, Ondrej Bojar, Silvie Cinková, Eva Fucíková, Marie Mikulová, Petr Pajas, Jan Popelka, et al. 2012. Announcing prague czech-english dependency treebank 2.0. In *LREC*, pages 3153–3160.

Chu-Ren Huang, Feng-Yi Chen, Keh-Jiann Chen, Zhao-ming Gao, and Kuang-Yu Chen. 2000. Sinica treebank: design criteria, annotation guidelines, and on-line interface. In *Proceedings of the second workshop on Chinese language processing: held in conjunction with the 38th ACL-Volume 12*, pages 29–37.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *16th Nordic Conference of Computational Linguistics*, pages 105–112. University of Tartu.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of COLING-ACL 2006*, pages 609–616. Association for Computational Linguistics.

David M Magerman. 1994. Natural language parsing as statistical pattern recognition. *arXiv preprint cmp-lg/9405009*.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT-EMNLP 2005*, pages 523–530.

Fandong Meng, Jun Xie, Linfeng Song, Yajuan Lü, and Qun Liu. 2013. Translation with source constituency and dependency trees. In *Proceedings of EMNLP 2010*, pages 1066–1076.

Joakim Nivre. 2006. *Inductive dependency parsing*. Springer.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT-NAACL 2007*, pages 404–411.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, volume 59.

Owen Rambow, Cassandre Creswell, Rachel Szekely, Harriet Taber, and Marilyn A Walker. 2002. A dependency treebank for english. In *Proceedings of LREC 2002*.

Owen Rambow. 2010. The simple truth about dependency and phrase structure representations: An opinion piece. In *Proceedings of HLT-NAACL 2010*, pages 337–340.

Honglin Sun and Daniel Jurafsky. 2004. Shallow semantc parsing of Chinese. In *Proceedings of HLT-NAACL 2004*, pages 249–256.

Weiwei Sun and Xiaojun Wan. 2013. Data-driven, PCFG-based and pseudo-PCFG-based models for Chinese dependency parsing. *Transactions of the Association for Computational Linguistics*, 1(1):301–314.

Heike Telljohann, Erhard W Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2003. Stylebook for the tübingen treebank of written german (tüba-d/z). In *Seminar für Sprachwissenschaft, Universität Tübingen, Germany*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL 2003-Volume 1*, pages 173–180.

Zhiguo Wang and Chengqing Zong. 2010. Phrase structure parsing with dependency structure. In *Proceedings of Coling 2010: Posters*, pages 1292–1300.

Fei Xia and Martha Palmer. 2001. Converting dependency structures to phrase structures. In *Proceedings of HLT 2001*, pages 1–5. Association for Computational Linguistics.

Fei Xia, Owen Rambow, Rajesh Bhatt, Martha Palmer, and Dipti Misra Sharma. 2009. Towards a multi-representational treebank. In *The 7th International Workshop on Treebanks and Linguistic Theories*.

Jun Xie, Haitao Mi, and Qun Liu. 2011. A novel dependency-to-string model for statistical machine translation. In *Proceedings of EMNLP 2011*, pages 216–226.

Nianwen Xue, Fei Xia, Shizhe Huang, and Anthony Kroch. 2000. The bracketing guidelines for the penn chinese treebank (3.0).

Nianwen Xue. 2007. Tapping the implicit information for the PS to DS conversion of the chinese treebank. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistics Theories*.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT 2003*, volume 3.

Shiwen Yu, Huiming Duan, Xuefeng Zhu, Bin Swen, and Baobao Chang. 2003. Specification for corpus processing at peking university: Word segmentation, pos tagging and phonetic notation. *Journal of Chinese Language and Computing*, 13(2):121–158.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of EMNLP 2008*, pages 562–571.

Yue Zhang and Stephen Clark. 2009. Transition-based parsing of the Chinese treebank using a global discriminative model. In *Proceedings of IWPT 2009*, pages 162–171.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL-HLT 2011: short papers-Volume 2*, pages 188–193.

Yue Zhang and Joakim Nivre. 2012. Analyzing the effect of global learning and beam-search on transition-based dependency parsing. In *COLING (Posters)*, pages 1391–1400.

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *Proceedings of ACL 2013*, pages 434–443.

Dexi Zhu. 1982. *Grammar Finder*. Commercial Press.