# Exploiting Multiple Treebanks for Parsing with Quasi-synchronous Grammars

**Zhenghua Li, Ting Liu,[*] Wanxiang Che**
Research Center for Social Computing and Information Retrieval
School of Computer Science and Technology
Harbin Institute of Technology, China
{lzh,tliu,car}@ir.hit.edu.cn

## Abstract

We present a simple and effective framework for exploiting multiple monolingual treebanks with different annotation guidelines for parsing. Several types of *transformation patterns* (TP) are designed to capture the systematic annotation inconsistencies among different treebanks. Based on such TPs, we design *quasi-synchronous grammar* features to augment the baseline parsing models. Our approach can significantly advance the state-of-the-art parsing accuracy on two widely used target treebanks (Penn Chinese Treebank 5.1 and 6.0) using the Chinese Dependency Treebank as the source treebank. The improvements are respectively 1.37% and 1.10% with automatic part-of-speech tags. Moreover, an indirect comparison indicates that our approach also outperforms previous work based on treebank conversion.

## 1 Introduction

The scale of available labeled data significantly affects the performance of statistical data-driven models. As a structural classification problem that is more challenging than binary classification and sequence labeling problems, syntactic parsing is more prone to suffer from the data sparseness problem. However, the heavy cost of treebanking typically limits one single treebank in both scale and genre. At present, learning from one single treebank seems inadequate for further boosting parsing accuracy.[1] Therefore, studies have recently resorted to other resources for the enhancement of parsing models, such as large-scale unlabeled data (Koo et al., 2008; Chen et al., 2009; Bansal and Klein, 2011; Zhou et al., 2011), and bilingual texts or cross-lingual treebanks (Burkett and Klein, 2008; Huang et al., 2009; Burkett et al., 2010; Chen et al., 2010).

| Treebanks | # of Words | Grammar |
|---|---|---|
| CTB5 | 0.51 million | Phrase structure |
| CTB6 | 0.78 million | Phrase structure |
| CDT | 1.11 million | Dependency structure |
| Sinica | 0.36 million | Phrase structure |
| TCT | about 1 million | Phrase structure |

Table 1: Several publicly available Chinese treebanks.

The existence of multiple monolingual treebanks opens another door for this issue. For example, table 1 lists a few publicly available Chinese treebanks that are motivated by different linguistic theories or applications. In the current paper, we utilize the first three treebanks, i.e., the Chinese Penn Treebank 5.1 (CTB5) and 6.0 (CTB6) (Xue et al., 2005), and the Chinese Dependency Treebank (CDT) (Liu et al., 2006). The Sinica treebank (Chen et al., 2003) and the Tsinghua Chinese Treebank (TCT) (Qiang, 2004) can be similarly exploited with our proposed approach, which we leave as future work.

Despite the divergence of annotation philosophy, these treebanks contain rich human knowledge on the Chinese syntax, thereby having a great deal of common ground. Therefore, exploiting multiple treebanks is very attractive for boosting parsing accuracy. Figure 1 gives an example with different an-

[*] Correspondence author: tliu@ir.hit.edu.cn

[1] Incorporating an increased number of global features, such as third-order features in graph-based parsers, slightly affects parsing accuracy (Koo and Collins, 2010; Li et al., 2011).

*Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 675–684,
Jeju, Republic of Korea, 8-14 July 2012.

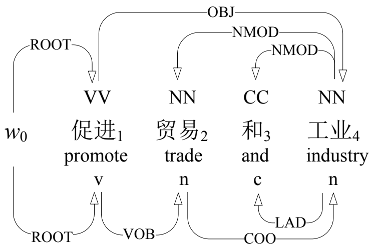

Figure 1: Example with annotations from CTB5 (upper) and CDT (under).

notations from CTB5 and CDT.[2] This example illustrates that the two treebanks annotate coordination constructions differently. In CTB5, the last noun is the head, whereas the first noun is the head in CDT.

One natural idea for multiple treebank exploitation is *treebank conversion*. First, the annotations in the source treebank are converted into the style of the target treebank. Then, both the converted treebank and the target treebank are combined. Finally, the combined treebank are used to train a better parser. However, the inconsistencies among different treebanks are normally nontrivial, which makes rule-based conversion infeasible. For example, a number of inconsistencies between CTB5 and CDT are lexicon-sensitive, that is, they adopt different annotations for some particular lexicons (or word senses). Niu et al. (2009) use sophisticated strategies to reduce the noises of the converted treebank after automatic treebank conversion.

The present paper proposes a simple and effective framework for this problem. The proposed framework avoids directly addressing the difficult annotation transformation problem, but focuses on modeling the annotation inconsistencies using *transformation patterns* (TP). The TPs are used to compose *quasi-synchronous grammar* (QG) features, such that the knowledge of the source treebank can inspire the target parser to build better trees. We conduct extensive experiments using CDT as the source treebank to enhance two target treebanks (CTB5 and CTB6). Results show that our approach can significantly boost state-of-the-art parsing accuracy. Moreover, an indirect comparison indicates that our approach also outperforms the treebank conversion approach of Niu et al. (2009).

## 2 Related Work

The present work is primarily inspired by Jiang et al. (2009) and Smith and Eisner (2009). Jiang et al. (2009) improve the performance of word segmentation and part-of-speech (POS) tagging on CTB5 using another large-scale corpus of different annotation standards (People's Daily). Their framework is similar to ours. However, handling syntactic annotation inconsistencies is significantly more challenging in our case of parsing. Smith and Eisner (2009) propose effective QG features for parser adaptation and projection. The first part of their work is closely connected with our work, but with a few important differences. First, they conduct simulated experiments on one treebank by manually creating a few trivial annotation inconsistencies based on two heuristic rules. They then focus on better adapting a parser to a new annotation style with few sentences of the target style. In contrast, we experiment with two real large-scale treebanks, and boost the state-of-the-art parsing accuracy using QG features. Second, we explore much richer QG features to fully exploit the knowledge of the source treebank. These features are tailored to the dependency parsing problem. In summary, the present work makes substantial progress in modeling structural annotation inconsistencies with QG features for parsing.

Previous work on treebank conversion primarily focuses on converting one grammar formalism of a treebank into another and then conducting a study on the converted treebank (Collins et al., 1999; Xia et al., 2008). The work by Niu et al. (2009) is, to our knowledge, the only study to date that combines the converted treebank with the existing target treebank. They automatically convert the dependency-structure CDT into the phrase-structure style of CTB5 using a statistical constituency parser trained on CTB5. Their experiments show that the combined treebank can significantly improve the performance of constituency parsers. However, their method requires several sophisticated strategies, such as corpus weighting and score interpolation, to reduce the influence of conversion errors. Instead of using the noisy converted treebank as additional training data, our approach allows the QG-

[2] CTB5 is converted to dependency structures following the standard practice of dependency parsing (Zhang and Clark, 2008b). Notably, converting a phrase-structure tree into its dependency-structure counterpart is straightforward and can be performed by applying heuristic head-finding rules.

enhanced parsing models to softly learn the systematic inconsistencies based on QG features, making our approach simpler and more robust.

Our approach is also intuitively related to *stacked learning* (SL), a machine learning framework that has recently been applied to dependency parsing to integrate two main-stream parsing models, i.e., graph-based and transition-based models (Nivre and McDonald, 2008; Martins et al., 2008). However, the SL framework trains two parsers on the same treebank and therefore does not need to consider the problem of annotation inconsistencies.

## 3 Dependency Parsing

Given an input sentence $\mathbf{x} = w_0 w_1 ... w_n$ and its POS tag sequence $\mathbf{t} = t_0 t_1 ... t_n$, the goal of dependency parsing is to build a dependency tree as depicted in Figure 1, denoted by $\mathbf{d} = \{(h, m, l) : 0 \leq h \leq n, 0 < m \leq n, l \in \mathcal{L}\}$, where $(h, m, l)$ indicates an directed arc from the *head word* (also called *father*) $w_h$ to the *modifier* (also called *child* or *dependent*) $w_m$ with a dependency label $l$, and $\mathcal{L}$ is the label set. We omit the label $l$ because we focus on unlabeled dependency parsing in the present paper. The artificial node $w_0$, which always points to the root of the sentence, is used to simplify the formalizations.

In the current research, we adopt the graph-based parsing models for their state-of-the-art performance in a variety of languages.[3] Graph-based models view the problem as finding the highest scoring tree from a directed graph. To guarantee the efficiency of the decoding algorithms, the score of a dependency tree is factored into the scores of some small parts (subtrees).

$$\begin{aligned} Score_{bs}(\mathbf{x}, \mathbf{t}, \mathbf{d}) &= \mathbf{w}_{bs} \cdot \mathbf{f}_{bs}(\mathbf{x}, \mathbf{t}, \mathbf{d}) \\ &= \sum_{p \subseteq \mathbf{d}} \mathbf{w}_{part} \cdot \mathbf{f}_{part}(\mathbf{x}, \mathbf{t}, p) \end{aligned}$$

where $p$ is a scoring part which contains one or more dependencies of $\mathbf{d}$, and $\mathbf{f}_{bs}(.)$ denotes the *basic parsing features*, as opposed to the *QG features*. Figure 2 lists the scoring parts used in our work, where $g$, $h$, $m$, and $s$, are word indices.

We implement three parsing models of varying strengths in capturing features to better understand the effect of the proposed QG features.

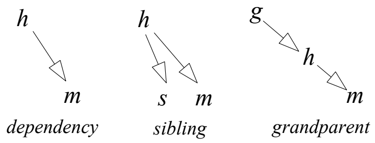

Figure 2: Scoring parts used in our graph-based parsing models.

- **The first-order model (O1)** only incorporates dependency parts (McDonald et al., 2005), and requires $O(n^3)$ parsing time.
- **The second-order model using only sibling parts (O2sib)** includes both dependency and sibling parts (McDonald and Pereira, 2006), and needs $O(n^3)$ parsing time.
- **The second-order model (O2)** uses all the scoring parts in Figure 2 (Koo and Collins, 2010). The time complexity of the decoding algorithm is $O(n^4)$.[4]

For the O2 model, the score function is rewritten as:

$$\begin{aligned} Score_{bs}(\mathbf{x}, \mathbf{t}, \mathbf{d}) = & \sum_{\{(h,m)\} \subseteq \mathbf{d}} \mathbf{w}_{dep} \cdot \mathbf{f}_{dep}(\mathbf{x}, \mathbf{t}, h, m) \\ & + \sum_{\{(h,s),(h,m)\} \subseteq \mathbf{d}} \mathbf{w}_{sib} \cdot \mathbf{f}_{sib}(\mathbf{x}, \mathbf{t}, h, s, m) \\ & + \sum_{\{(g,h),(h,m)\} \subseteq \mathbf{d}} \mathbf{w}_{grd} \cdot \mathbf{f}_{grd}(\mathbf{x}, \mathbf{t}, g, h, m) \end{aligned}$$

where $\mathbf{f}_{dep}(.)$, $\mathbf{f}_{sib}(.)$ and $\mathbf{f}_{grd}(.)$ correspond to the features for the three kinds of scoring parts. We adopt the standard features following Li et al. (2011). For the O1 and O2sib models, the above formula is modified by deactivating the extra parts.

## 4 Dependency Parsing with QG Features

Smith and Eisner (2006) propose the QG for machine translation (MT) problems, allowing greater syntactic divergences between the two languages. Given a source sentence $\mathbf{x}'$ and its syntactic tree $\mathbf{d}'$, a QG defines a monolingual grammar that generates translations of $\mathbf{x}'$, which can be denoted by $p(\mathbf{x}, \mathbf{d}, \mathbf{a}|\mathbf{x}', \mathbf{d}')$, where $\mathbf{x}$ and $\mathbf{d}$ refer to a translation and its parse, and $\mathbf{a}$ is a cross-language alignment. Under a QG, any portion of $\mathbf{d}$ can be aligned to any

[3] Our approach can equally be applied to transition-based parsing models (Yamada and Matsumoto, 2003; Nivre, 2003) with minor modifications.

[4] We use the coarse-to-fine strategy to prune the search space, which largely accelerates the decoding procedure (Koo and Collins, 2010).

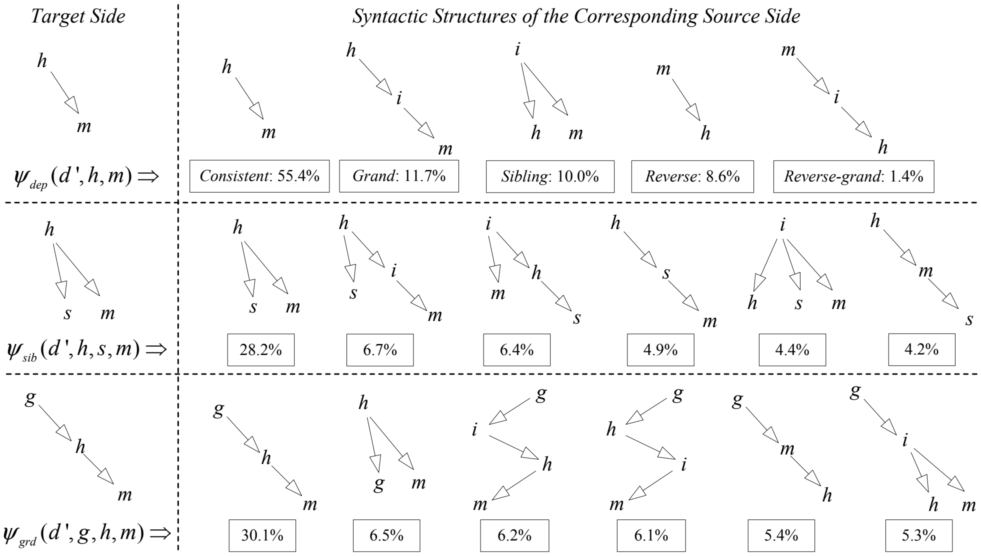
Target Side
Syntactic Structures of the Corresponding Source Side
$\psi_{dep}(d', h, m) \Rightarrow$
Consistent: 55.4%
Grand: 11.7%
Sibling: 10.0%
Reverse: 8.6%
Reverse-grand: 1.4%
$\psi_{sib}(d', h, s, m) \Rightarrow$
28.2%
6.7%
6.4%
4.9%
4.4%
4.2%
$\psi_{grd}(d', g, h, m) \Rightarrow$
30.1%
6.5%
6.2%
6.1%
5.4%
5.3%
h
m
s
i
g

Figure 4: Most frequent transformation patterns (TPs) when using CDT as the source treebank and CTB5 as the target. A TP comprises two syntactic structures, one in the source side and the other in the target side, and denotes the process by which the left-side subtree is transformed into the right-side structure. Functions $\psi_{dep}(.)$, $\psi_{sib}(.)$, and $\psi_{grd}(.)$ return the specific TP type for a candidate scoring part according to the source tree $\mathbf{d}'$.

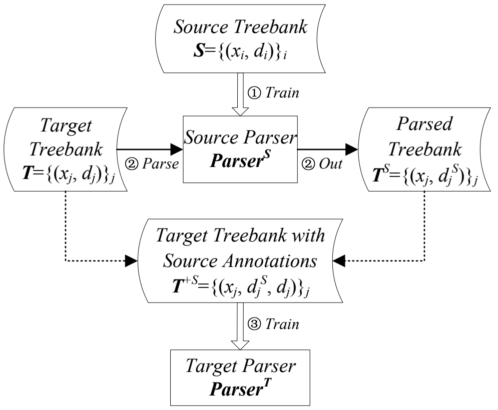
Source Treebank
$\boldsymbol{S}=\{(x_i, d_i)\}_i$
① Train
Target Treebank
$\boldsymbol{T}=\{(x_j, d_j)\}_j$
② Parse
Source Parser
$\boldsymbol{Parser^S}$
② Out
Parsed Treebank
$\boldsymbol{T}^S=\{(x_j, d_j^S)\}_j$
Target Treebank with Source Annotations
$\boldsymbol{T}^{+S}=\{(x_j, d_j^S, d_j)\}_j$
③ Train
Target Parser
$\boldsymbol{Parser^T}$

Figure 3: Framework of our approach.

portion of $\mathbf{d}'$, and the construction of $\mathbf{d}$ can be inspired by arbitrary substructures of $\mathbf{d}'$. To date, QGs have been successfully applied to various tasks, such as word alignment (Smith and Eisner, 2006), machine translation (Gimpel and Smith, 2011), question answering (Wang et al., 2007), and sentence simplification (Woodsend and Lapata, 2011).

In the present work, we utilize the idea of the QG for the exploitation of multiple monolingual treebanks. The key idea is to let the parse tree of one style inspire the parsing process of another style. Different from a MT process, our problem considers one single sentence ($\mathbf{x} = \mathbf{x}'$), and the alignment $\mathbf{a}$ is trivial. Figure 3 shows the framework of our approach. First, we train a statistical parser on the *source treebank*, which is called the *source parser*. The source parser is then used to parse the whole *target treebank*. At this point, the target treebank contains two sets of annotations, one conforming to the source style, and the other conforming to the target style. During both the training and test phases, the *target parser* are inspired by the source annotations, and the score of a target dependency tree becomes

$$\begin{aligned} Score(\mathbf{x}, \mathbf{t}, \mathbf{d}', \mathbf{d}) =& Score_{bs}(\mathbf{x}, \mathbf{t}, \mathbf{d}) \\ &+ Score_{qg}(\mathbf{x}, \mathbf{t}, \mathbf{d}', \mathbf{d}) \end{aligned}$$

The first part corresponds to the baseline model, whereas the second part is affected by the source tree $\mathbf{d}'$ and can be rewritten as

$$Score_{qg}(\mathbf{x}, \mathbf{t}, \mathbf{d}', \mathbf{d}) = \mathbf{w}_{qg} \cdot \mathbf{f}_{qg}(\mathbf{x}, \mathbf{t}, \mathbf{d}', \mathbf{d})$$

where $\mathbf{f}_{qg}(.)$ denotes the *QG features*. We expect the QG features to encourage or penalize certain scoring parts in the target side according to the source tree $\mathbf{d}'$. Taking Figure 1 as an example, suppose that the upper structure is the target. The target parser can raise the score of the candidate dependence *"and"* ← *"industry"*, because the dependency also appears in the source structure, and evidence in the training data shows that both annotation styles handle conjunctions in the same manner. Similarly, the parser may add weight to *"trade"* ← *"industry"*, considering that the *reverse* arc is in the source structure. Therefore, the QG-enhanced model must learn the systematic consistencies and inconsistencies from the training data.

To model such consistency or inconsistency systematicness, we propose the use of TPs for encoding the structural correspondence between the source and target styles. Figure 4 presents the three kinds of TPs used in our model, which correspond to the three scoring parts of our parsing models.

Dependency TPs shown in the first row consider how one dependency in the target side is transformed in the source annotations. We only consider the five cases shown in the figure. The percentages in the lower boxes refer to the proportion of the corresponding pattern, which are counted from the training data of the target treebank with source annotations $T^{+S}$. We can see that the noisy source structures and the gold-standard target structures have 55.4% common dependencies. If the source structure does not belong to any of the listed five cases, $\psi_{dep}(\mathbf{d}', h, m)$ returns *"else"* (12.9%). We could consider more complex structures, such as $h$ being the grand grand father of $m$, but statistics show that more complex transformations become very scarce in the training data.

For the reason that dependency TPs can only model how one dependency in the target structure is transformed, we consider more complex transformations for the other two kinds of scoring parts of the target parser, i.e., the sibling and grand TPs shown in the bottom two rows. We only use high-frequency TPs of a proportion larger than 1.0%, aggregate others as *"else"*, which leaves us with 21 sibling TPs and 22 grand TPs.

Based on these TPs, we propose the QG features for enhancing the baseline parsing models, which are shown in Table 2. The type of the TP is conjoined with the related words and POS tags, such that the QG-enhanced parsing models can make more elaborate decisions based on the context. Then, the score contributed by the QG features can be redefined as

$$\begin{aligned}
&Score_{qg}(\mathbf{x}, \mathbf{t}, \mathbf{d}', \mathbf{d}) = \\
&\quad \sum_{\{(h,m)\}\subseteq\mathbf{d}} \mathbf{w}_{qg\text{-}dep} \cdot \mathbf{f}_{qg\text{-}dep}(\mathbf{x}, \mathbf{t}, \mathbf{d}', h, m) \\
&+ \sum_{\{(h,s),(h,m)\}\subseteq\mathbf{d}} \mathbf{w}_{qg\text{-}sib} \cdot \mathbf{f}_{qg\text{-}sib}(\mathbf{x}, \mathbf{t}, \mathbf{d}', h, s, m) \\
&+ \sum_{\{(g,h),(h,m)\}\subseteq\mathbf{d}} \mathbf{w}_{qg\text{-}grd} \cdot \mathbf{f}_{qg\text{-}grd}(\mathbf{x}, \mathbf{t}, \mathbf{d}', g, h, m)
\end{aligned}$$

which resembles the baseline model and can be naturally handled by the decoding algorithms.

## 5 Experiments and Analysis

We use the CDT as the source treebank (Liu et al., 2006). CDT consists of 60,000 sentences from the People's Daily in 1990s. For the target treebank, we use two widely used versions of Penn Chinese Treebank, i.e., CTB5 and CTB6, which consist of Xinhua newswire, Hong Kong news and articles from Sinarama news magazine (Xue et al., 2005). To facilitate comparison with previous results, we follow Zhang and Clark (2008b) for data split and constituency-to-dependency conversion of CTB5. CTB6 is used as the Chinese data set in the CoNLL 2009 shared task (Hajič et al., 2009). Therefore, we adopt the same setting.

CDT and CTB5/6 adopt different POS tag sets, and converting from one tag set to another is difficult (Niu et al., 2009).[5] To overcome this problem, we use the People's Daily corpus (PD),[6] a large-scale corpus annotated with word segmentation and POS tags, to train a statistical POS tagger. The tagger produces a universal layer of POS tags for both the source and target treebanks. Based on the common tags, the source parser projects the source annotations into the target treebanks. PD comprises approximately 300 thousand sentences of with approximately 7 million words from the first half of 1998 of People's Daily.

Table 3 summarizes the data sets used in the present work. CTB5X is the same with CTB5 but follows the data split of Niu et al. (2009). We use CTB5X to compare our approach with their treebank conversion method (see Table 9).

[5] The word segmentation standards of the two treebanks also slightly differs, which are not considered in this work.

[6] http://icl.pku.edu.cn/icl_groups/corpustagging.asp

| $\mathbf{f}_{qg\text{-}dep}(\mathbf{x}, \mathbf{t}, \mathbf{d}', h, m)$<br>$\oplus dir(h, m) \circ dist(h, m)$ | $\mathbf{f}_{qg\text{-}sib}(\mathbf{x}, \mathbf{t}, \mathbf{d}', h, s, m)$<br>$\oplus dir(h, m)$ | $\mathbf{f}_{qg\text{-}grd}(\mathbf{x}, \mathbf{t}, \mathbf{d}', g, h, m)$<br>$\oplus dir(h, m) \circ dir(g, h)$ |
|---|---|---|
| $\psi_{dep}(\mathbf{d}', h, m) \circ t_h \circ t_m$ | $\psi_{sib}(\mathbf{d}', h, s, m) \circ t_h \circ t_s \circ t_m$ | $\psi_{grd}(\mathbf{d}', g, h, m) \circ t_g \circ t_h \circ t_m$ |
| $\psi_{dep}(\mathbf{d}', h, m) \circ w_h \circ t_m$ | $\psi_{sib}(\mathbf{d}', h, s, m) \circ w_h \circ t_s \circ t_m$ | $\psi_{grd}(\mathbf{d}', g, h, m) \circ w_g \circ t_h \circ t_m$ |
| $\psi_{dep}(\mathbf{d}', h, m) \circ t_h \circ w_m$ | $\psi_{sib}(\mathbf{d}', h, s, m) \circ t_h \circ w_s \circ t_m$ | $\psi_{grd}(\mathbf{d}', g, h, m) \circ t_g \circ w_h \circ t_m$ |
| $\psi_{dep}(\mathbf{d}', h, m) \circ w_h \circ w_m$ | $\psi_{sib}(\mathbf{d}', h, s, m) \circ t_h \circ t_s \circ w_m$ | $\psi_{grd}(\mathbf{d}', g, h, m) \circ t_g \circ t_h \circ w_m$ |
|  | $\psi_{sib}(\mathbf{d}', h, s, m) \circ t_s \circ t_m$ | $\psi_{grd}(\mathbf{d}', g, h, m) \circ t_g \circ t_m$ |

Table 2: QG features used to enhance the baseline parsing models. $dir(h, m)$ denotes the direction of the dependency $(h, m)$, whereas $dist(h, m)$ is the distance $|h - m|$. $\oplus dir(h, m) \circ dist(h, m)$ indicates that the features listed in the corresponding column are also conjoined with $dir(h, m) \circ dist(h, m)$ to form new features.

| Corpus | Train | Dev | Test |
|---|---|---|---|
| PD | 281,311 | 5,000 | 10,000 |
| CDT | 55,500 | 1,500 | 3,000 |
| CTB5 | 16,091 | 803 | 1,910 |
| CTB5X | 18,104 | 352 | 348 |
| CTB6 | 22,277 | 1,762 | 2,556 |

Table 3: Data used in this work (in sentence number).

We adopt *unlabeled attachment score* (UAS) as the primary evaluation metric. We also use *Root accuracy* (RA) and *complete match rate* (CM) to give more insights. All metrics exclude punctuation. We adopt Dan Bikel's randomized parsing evaluation comparator for significance test (Noreen, 1989).[7]

For all models used in current work (POS tagging and parsing), we adopt averaged perceptron to train the feature weights (Collins, 2002). We train each model for 10 iterations and select the parameters that perform best on the development set.

### 5.1 Preliminaries

This subsection describes how we project the source annotations into the target treebanks. First, we train a statistical POS tagger on the training set of PD, which we name $Tagger^{PD}$.[8] The tagging accuracy on the test set of PD is 98.30%.

We then use $Tagger^{PD}$ to produce POS tags for all the treebanks (CDT, CTB5, and CTB6).

Based on the common POS tags, we train a second-order source parser (O2) on CDT, denoted by $Parser^{CDT}$. The UAS on CDT-test is 84.45%. We then use $Parser^{CDT}$ to parse CTB5 and CTB6. At this point, both CTB5 and CTB6 contain dependency structures conforming to the style of CDT.

| Models | without QG | with QG |
|---|---|---|
| O2 | 86.13 | 86.44 (+0.31, $p = 0.06$) |
| O2sib | 85.63 | 86.17 (+0.54, $p = 0.003$) |
| O1 | 83.16 | 84.40 (+1.24, $p < 10^{-5}$) |
| Li11 | 86.18 | — |
| Z&N11 | 86.00 | — |

Table 4: Parsing accuracy (UAS) comparison on CTB5-test with gold-standard POS tags. Li11 refers to the second-order graph-based model of Li et al. (2011), whereas Z&N11 is the feature-rich transition-based model of Zhang and Nivre (2011).

### 5.2 CTB5 as the Target Treebank

Table 4 shows the results when the gold-standard POS tags of CTB5 are adopted by the parsing models. We aim to analyze the efficacy of QG features under the ideal scenario wherein the parsing models suffer from no error propagation of POS tagging. We determine that our baseline O2 model achieves comparable accuracy with the state-of-the-art parsers. We also find that QG features can boost the parsing accuracy by a large margin when the baseline parser is weak (O1). The improvement shrinks for stronger baselines (O2sib and O2). This phenomenon is understandable. When gold-standard POS tags are available, the baseline features are very reliable and the QG features becomes less helpful for more complex models. The p-values in parentheses present the statistical significance of the improvements.

We then turn to the more realistic scenario wherein the gold-standard POS tags of the target treebank are unavailable. We train a POS tagger on the training set of CTB5 to produce the automatic

[7] `http://www.cis.upenn.edu/[normal-wave~]dbikel/software.html`

[8] We adopt the Chinese-oriented POS tagging features proposed in Zhang and Clark (2008a).

| Models | without QG | with QG |
|---|---|---|
| O2 | 79.67 | 81.04 (+1.37) |
| O2sib | 79.25 | 80.45 (+1.20) |
| O1 | 76.73 | 79.04 (+2.31) |
| Li11 joint | 80.79 | — |
| Li11 pipeline | 79.29 | — |

Table 5: Parsing accuracy (UAS) comparison on CTB5-test with automatic POS tags. The improvements shown in parentheses are all statistically significant ($p < 10^{-5}$).

| Setting | UAS | CM | RA |
|---|---|---|---|
| $\mathbf{f}_{bs}(.)$ | 79.67 | 26.81 | 73.82 |
| $\mathbf{f}_{qg}(.)$ | 79.15 | 26.34 | 74.71 |
| $\mathbf{f}_{bs}(.)$ + $\mathbf{f}_{qg}(.)$ | 81.04 | 29.63 | 77.17 |
| $\mathbf{f}_{bs}(.)$ + $\mathbf{f}_{qg\text{-}dep}(.)$ | 80.82 | 28.80 | 76.28 |
| $\mathbf{f}_{bs}(.)$ + $\mathbf{f}_{qg\text{-}sib}(.)$ | 80.86 | 28.48 | 76.18 |
| $\mathbf{f}_{bs}(.)$ + $\mathbf{f}_{qg\text{-}grd}(.)$ | 80.88 | 28.90 | 76.34 |

Table 6: Feature ablation for Parser-O2 on CTB5-test with automatic POS tags.

POS tags for the development and test sets of CTB5. The tagging accuracy is 93.88% on the test set. The automatic POS tags of the training set are produced using 10-fold cross-validation.[9]

Table 5 shows the results. We find that QG features result in a surprisingly large improvement over the O1 baseline and can also boost the state-of-the-art parsing accuracy by a large margin. Li et al. (2011) show that a joint POS tagging and dependency parsing model can significantly improve parsing accuracy over a pipeline model. Our QG-enhanced parser outperforms their best joint model by 0.25%. Moreover, the QG features can be used to enhance a joint model and achieve higher accuracy, which we leave as future work.

### 5.3 Analysis Using Parser-O2 with AUTO-POS

We then try to gain more insights into the effect of the QG features through detailed analysis. We select the state-of-the-art O2 parser and focus on the realistic scenario with automatic POS tags.

Table 6 compares the efficacy of different feature sets. The first major row analyzes the efficacy of the basic features $\mathbf{f}_{bs}(.)$ and the QG features $\mathbf{f}_{qg}(.)$. When using the few QG features in Table 2, the accuracy is very close to that when using the basic features. Moreover, using both features generates a large improvement. The second major row compares the efficacy of the three kinds of QG features corresponding to the three types of scoring parts. We can see that the three feature sets are similarly effective and yield comparable accuracies. Combining these features generate an additional improvement of approximately 0.2%. These results again demonstrate that all the proposed QG features are effective.

Figure 5 describes how the performance varies when the scale of CTB5 and CDT changes. In the left subfigure, the parsers are trained on part of the CTB5-train, and "16" indicates the use of all the training instances. Meanwhile, the source parser $Parser^{CDT}$ is trained on the whole CDT-train. We can see that QG features render larger improvement when the target treebank is of smaller scale, which is quite reasonable. More importantly, the curves indicate that **a QG-enhanced parser trained on a target treebank of 16,000 sentences may achieve comparable accuracy with a baseline parser trained on a treebank that is double the size (32,000)**, which is very encouraging.

In the right subfigure, the target treebank is trained on the whole CTB5-train, whereas the source parser is trained on part of the CDT-train, and "55.5" indicates the use of all. The curve clearly demonstrates that the QG features are more helpful when the source treebank gets larger, which can be explained as follows. A larger source treebank can teach a source parser of higher accuracy; then, the better source parser can parse the target treebank more reliably; and finally, the target parser can better learn the annotation divergences based on QG features. These results demonstrate the effectiveness and stability of our approach.

Table 7 presents the detailed effect of the QG features on different dependency patterns. A pattern "VV → NN" refers to a right-directed dependency with the head tagged as "VV" and the modifier tagged as "NN". whereas "←" means left-directed. The "w/o QG" column shows the number of the corresponding dependency pattern that appears in the gold-standard trees but misses in the results of the baseline parser, whereas the signed figures in the "+QG" column are the changes made by the QG-

[9] We could use the POS tags produced by $Tagger^{PD}$ in Section 5.1, which however would make it difficult to compare our results with previous ones. Moreover, inferior results may be gained due to the differences between CTB5 and PD in word segmentation standards and text sources.

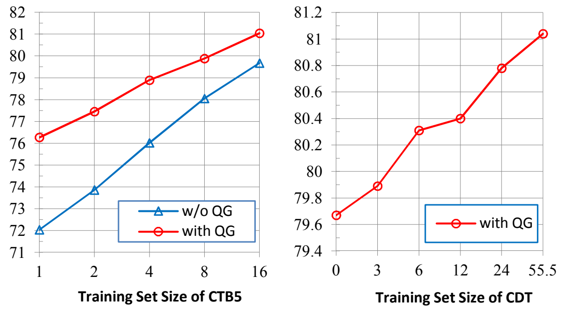

Figure 5: Parsing accuracy (UAS) comparison on CTB5-test when the scale of CDT and CTB5 varies (thousands in sentence number).

| Dependency | w/o QG | +QG | Descriptions |
|---|---|---|---|
| NN ← NN | 858 | -78 | noun modifier or coordinating nouns |
| VV → VV | 777 | -41 | object clause or coordinating verbs |
| VV ← VV | 570 | -38 | subject clause |
| VV → NN | 509 | -79 | verb and its object |
| $w_0$ → VV | 357 | -57 | verb as sentence root |
| VV ← NN | 328 | -32 | attributive clause |
| P ← VV | 278 | -37 | preposition phrase attachment |
| VV → DEC | 233 | -33 | attributive clause and auxiliary DE |
| P → NN | 175 | -35 | preposition and its object |

Table 7: Detailed effect of QG features on different dependency patterns.

enhanced parser. We only list the patterns with an absolute change larger than 30. We find that the QG features can significantly help a variety of dependency patterns (i.e., reducing the missing number).

### 5.4 CTB6 as the Target Treebank

We use CTB6 as the target treebank to further verify the efficacy of our approach. Compared with CTB5, CTB6 is of larger scale and is converted into dependency structures according to finer-grained head-finding rules (Hajič et al., 2009). We directly adopt the same transformation patterns and features tuned on CTB5. Table 8 shows results. The improvements are similar to those on CTB5, demonstrating that our approach is effective and robust. We list the top three systems of the CoNLL 2009 shared task in Table 8, showing that our approach also advances the state-of-the-art parsing accuracy on this data set.[10]

| Models | without QG | with QG |
|---|---|---|
| O2 | 83.23 | 84.33 (+1.10) |
| O2sib | 82.87 | 84.11 (+1.37) |
| O1 | 80.29 | 82.76 (+2.47) |
| Bohnet (2009) | 82.68 | — |
| Che et al. (2009) | 82.11 | — |
| Gesmundo et al. (2009) | 81.70 | — |

Table 8: Parsing accuracy (UAS) comparison on CTB6-test with automatic POS tags. The improvements shown in parentheses are all statistically significant ($p < 10^{-5}$).

| Models | baseline | with another treebank |
|---|---|---|
| Ours | 84.16 | 86.67 (+2.51) |
| GP (Niu et al., 2009) | 82.42 | 84.06 (+1.64) |

Table 9: Parsing accuracy (UAS) comparison on the test set of CTB5X. Niu et al. (2009) use the maximum entropy inspired generative parser (GP) of Charniak (2000) as their constituent parser.

### 5.5 Comparison with Treebank Conversion

As discussed in Section 2, Niu et al. (2009) automatically convert the dependency-structure CDT to the phrase-structure annotation style of CTB5X and use the converted treebank as additional labeled data. We convert their phrase-structure results on CTB5X-test into dependency structures using the same head-finding rules. To compare with their results, we run our baseline and QG-enhanced O2 parsers on CTB5X. Table 9 presents the results.[11] The indirect comparison indicates that our approach can achieve larger improvement than their treebank conversion based method.

## 6 Conclusions

The current paper proposes a simple and effective framework for exploiting multiple large-scale treebanks of different annotation styles. We design rich TPs to model the annotation inconsistencies and consequently propose QG features based on these TPs. Extensive experiments show that our approach can effectively utilize the syntactic knowledge from another treebank and significantly improve the state-of-the-art parsing accuracy.

[10] We reproduce their UASs using the data released by the organizer: http://ufal.mff.cuni.cz/conll2009-st/results/results.php. The parsing accuracies of the top systems may be underestimated since the accuracy of the provided POS tags in CoNLL 2009 is only 92.38% on the test set, while the POS tagger used in our experiments reaches 94.08%.

[11] We thank the authors for sharing their results. Niu et al. (2009) also use the reranker (RP) of Charniak and Johnson (2005) as a stronger baseline, but the results are missing. They find a less improvement on F score with RP than with GP (0.9% vs. 1.1%). We refer to their Table 5 and 6 for details.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 61133012, the National "863" Major Projects via grant 2011AA01A207, and the National "863" Leading Technology Research Project via grant 2012AA011102.

## References

Mohit Bansal and Dan Klein. 2011. Web-scale features for full-scale parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 693–702, Portland, Oregon, USA, June. Association for Computational Linguistics.

Bernd Bohnet. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 67–72, Boulder, Colorado, June. Association for Computational Linguistics.

David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 877–886, Honolulu, Hawaii, October. Association for Computational Linguistics.

David Burkett, Slav Petrov, John Blitzer, and Dan Klein. 2010. Learning better monolingual models with unannotated bilingual text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 46–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL-05*, pages 173–180.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *ANLP'00*, pages 132–139.

Wanxiang Che, Zhenghua Li, Yongqiang Li, Yuhang Guo, Bing Qin, and Ting Liu. 2009. Multilingual dependency-based syntactic and semantic parsing. In *Proceedings of CoNLL 2009: Shared Task*, pages 49–54.

Keh-Jiann Chen, Chi-Ching Luo, Ming-Chung Chang, Feng-Yi Chen, Chao-Jan Chen, Chu-Ren Huang, and Zhao-Ming Gao, 2003. *Sinica treebank: Design criteria,representational issues and implementation*, chapter 13, pages 231–248. Kluwer Academic Publishers.

Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 570–579, Singapore, August. Association for Computational Linguistics.

Wenliang Chen, Jun'ichi Kazama, and Kentaro Torisawa. 2010. Bitext dependency parsing with bilingual subtree constraints. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 21–29, Uppsala, Sweden, July. Association for Computational Linguistics.

Micheal Collins, Lance Ramshaw, Jan Hajic, and Christoph Tillmann. 1999. A statistical parser for czech. In *ACL 1999*, pages 505–512.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*.

Andrea Gesmundo, James Henderson, Paola Merlo, and Ivan Titov. 2009. A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *Proceedings of CoNLL 2009: Shared Task*, pages 37–42.

Kevin Gimpel and Noah A. Smith. 2011. Quasi-synchronous phrase dependency grammars for machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 474–485, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL 2009*.

Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1222–1231, Singapore, August. Association for Computational Linguistics.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging – a case study. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 522–530, Suntec, Singapore, August. Association for Computational Linguistics.

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1–11, Uppsala, Sweden, July. Association for Computational Linguistics.

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

Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for chinese pos tagging and dependency parsing. In *EMNLP 2011*, pages 1180–1191.

Ting Liu, Jinshan Ma, and Sheng Li. 2006. Building a dependency treebank for improving Chinese parser. In *Journal of Chinese Language and Computing*, volume 16, pages 207–224.

Andr— F. T. Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *EMNLP'08*, pages 157–166.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL 2006*.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL 2005*, pages 91–98.

Zheng-Yu Niu, Haifeng Wang, and Hua Wu. 2009. Exploiting heterogeneous treebanks for parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 46–54, Suntec, Singapore, August. Association for Computational Linguistics.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL 2008*, pages 950–958.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.

Eric W. Noreen. 1989. *Computer-intensive methods for testing hypotheses: An introduction*. John Wiley & Sons, Inc., New York. Book (ISBN 0471611360 ).

Zhou Qiang. 2004. Annotation scheme for chinese treebank. *Journal of Chinese Information Processing*, 18(4):1–8.

David Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 23–30, New York City, June. Association for Computational Linguistics.

David A. Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 822–831, Singapore, August. Association for Computational Linguistics.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Fei Xia, Rajesh Bhatt, Owen Rambow, Martha Palmer, and Dipti Misra. Sharma. 2008. Towards a multi-representational treebank. In *In Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories*.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. In *Natural Language Engineering*, volume 11, pages 207–238.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT 2003*, pages 195–206.

Yue Zhang and Stephen Clark. 2008a. Joint word segmentation and POS tagging using a single perceptron. In *Proceedings of ACL-08: HLT*, pages 888–896.

Yue Zhang and Stephen Clark. 2008b. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 562–571, Honolulu, Hawaii, October. Association for Computational Linguistics.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA, June. Association for Computational Linguistics.

Guangyou Zhou, Jun Zhao, Kang Liu, and Li Cai. 2011. Exploiting web-derived selectional preference to improve statistical dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1556–1565, Portland, Oregon, USA, June. Association for Computational Linguistics.