Toward Tree Substitution Grammars with Latent Annotations

Francis Ferraro and **Benjamin Van Durme** and **Matt Post**
Center for Language and Speech Processing, and
Human Language Technology Center of Excellence
Johns Hopkins University

Abstract

We provide a model that extends the split-merge framework of Petrov et al. (2006) to jointly learn latent annotations and Tree Substitution Grammars (TSGs). We then conduct a variety of experiments with this model, first inducing grammars on a portion of the Penn Treebank and the Korean Treebank 2.0, and next experimenting with grammar refinement from a single nonterminal and from the Universal Part of Speech tagset. We present qualitative analysis showing promising signs across all experiments that our combined approach successfully provides for greater flexibility in grammar induction within the structured guidance provided by the treebank, leveraging the complementary natures of these two approaches.

1 Introduction

Context-free grammars (CFGs) are a useful tool for describing the structure of language, modeling a variety of linguistic phenomena while still permitting efficient inference. However, it is widely acknowledged that CFGs employed in practice make unrealistic independence and structural assumptions, resulting in grammars that are overly permissive. One successful approach has been to refine the nonterminals of grammars, first manually (Johnson, 1998; Klein and Manning, 2003) and later automatically (Matsuzaki et al., 2005; Dreyer and Eisner, 2006; Petrov et al., 2006). In addition to improving parsing accuracy, the automatically learned *latent annotations* of these latter approaches yield results that accord well with human intuitions, especially at the lexical or preterminal level (for example, separating demonstrative adjectives from definite articles under the DT tag). It is more difficult, though, to extend this analysis to higher-level nonterminals, where the long-distance interactions among latent annotations of internal nodes are subtle and difficult to trace.

In another line of work, many researchers have examined the use of formalisms with an *extended domain of locality* (Joshi and Schabes, 1997), where the basic grammatical units are arbitrary tree fragments instead of traditional depth-one context-free grammar productions. In particular, Tree Substitution Grammars (TSGs) retain the context-free properties of CFGs (and thus the cubic-time inference) while at the same time allowing for the modeling of long distance dependencies. Fragments from such grammars are intuitive, capturing exactly the sorts of phrasal-level properties (such as predicate-argument structure) that are not present in Treebank CFGs and which are difficult to model with latent annotations.

This paper is motivated by the complementarity of these approaches. We present our progress in learning latent-variable TSGs in a joint approach that extends the split-merge framework of Petrov et al. (2006). We present our current results on the Penn and Korean treebanks (Marcus et al., 1993; Han et al., 2001), demonstrating that we are able to learn fragments that draw on the strengths of both approaches. Table 1 situates this work among other contributions.

In addition to experimenting directly with the Penn and Korean Treebanks, we also conducted two experiments in this framework with the Universal

NAACL-HLT Workshop on the Induction of Linguistic Structure, pages 23–30,
Montréal, Canada, June 3-8, 2012.

	CFG	TSG
none	Charniak '97	Cohn et al. '09
manual	Klein & Manning '03	Bansal & Klein '10
automatic	Matsuzaki et al. '05 Petrov et al. '06 Dreyer & Eisner '06	*This paper*

Table 1: Representative prior work in learning refinements for context-free and tree substitution grammars, with zero, manual, or automatically induced latent annotations.

POS tagset (Petrov et al., 2011). First, we investigate whether the tagset can be automatically derived after mapping all nonterminals to a single, coarse nonterminal. Second, we begin with the mapping defined by the tagset, and investigate how closely the learned annotations resemble the original treebank. Together with our TSG efforts, this work is aimed at increased flexibility in the grammar induction process, while retaining the use of Treebanks for structural guidance.

2 Background

2.1 Latent variable grammars

Latent annotation learning is motivated by the observed coarseness of the nonterminals in treebank grammars, which often group together nodes with different grammatical roles and distributions (such as the role of NPs in subject and object position). Johnson (1998) presented a simple parent-annotation scheme that resulted in significant parsing improvement. Klein and Manning (2003) built on these observations, introducing a series of manual refinements that captured multiple linguistic phenomena, leading to accurate and fast unlexicalized parsing. Later, automated methods for nonterminal refinement were introduced, first splitting all categories equally (Matsuzaki et al., 2005), and later refining nonterminals to different degrees (Petrov et al., 2006) in a split-merge EM framework. This latter approach was able to recover many of the splits manually determined by Klein and Manning (2003), while also discovering interesting, novel clusterings, especially at the lexical level. However, phrasal-level analysis of latent-variable grammars is more difficult. (2006) observed that these grammars could learn long-distance dependencies through sequences of substates that place all or most of their weight on

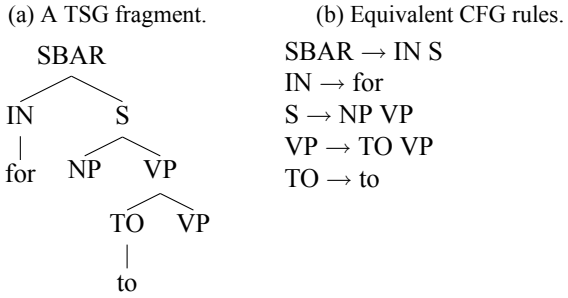


Figure 1: Simple example of a TSG fragment and an equivalent representation with a CFG.

particular productions, but such patterns must be discovered manually via extensive analysis.

2.2 Tree substitution grammars

Tree substitution grammars (TSGs) allow for complementary analysis. These grammars employ an *extended domain of locality* over traditional context-free grammars by generalizing the atomic units of the grammar from depth-one productions to fragments of arbitrary size. An example TSG fragment along with equivalent CFG rules are depicted in Figure 1. The two formalisms are weakly equivalent, and computing the most probable derivation of a sentence with a TSG can be done in cubic time.

Unfortunately, learning TSGs is not straightforward, in large part because TSG-specific resources (e.g., large scale TSG-annotated treebanks) do not exist. One class of existing approaches, known as Data-Oriented Parsing, simply uses all the fragments (Bod, 1993, DOP). This does not scale well to large treebanks, forcing the use of implicit representations (Goodman, 1996) or heuristic subsets (Bod, 2001). It has also been generally observed that the use of all fragments results in poor, overfit grammars, though this can be addressed with held-out data (Zollmann and Sima'an, 2005) or statistical estimators to rule out fragments that are unlikely to generalize (Zuidema, 2007). More recently, a number of groups have found success employing Bayesian non-parametric priors (Post and Gildea, 2009; Cohn et al., 2010), which put a downward pressure on fragment size except where the data warrant the inclusion of larger fragments. Unfortunately, proper inference under these models is intractable, and though Monte Carlo techniques can

provide an approximation, the samplers can be complex, difficult to code, and slow to converge.

This history suggests two approaches to state-split TSGs: (1) a Bayesian non-parametric sampling approach (incorporate state-splitting into existing TSG work), or (2) EM (incorporate TSG induction into existing state-splitting work). We choose the latter path, and in the next section will describe our approach which combines the simplicity of DOP, the intuitions motivating the Bayesian approach, and the efficiency of EM-based state-splitting.

In related work, Bansal and Klein (2010) combine (1996)'s implicit DOP representation with a number of the manual refinements described in Klein and Manning (2003). They achieve some of the best reported parsing scores for TSG work and demonstrate the complementarity of the tasks, but their approach is not able to learn arbitrary distributions over fragments, and the state splits are determined in a fixed pre-processing step. Our approach addresses both of these limitations.

3 State-Split TSG Induction

In this section we describe how we combine the ideas of dop, Bayesian-induced TSGs and Petrov et al. (2006)'s state-splitting framework.[1] We are able to do so by adding a **coupling** step to each iteration. That is, each iteration is of the form:

(1) **split** all symbols in two,

(2) **merge** 50% of the splits, and

(3) **couple** existing fragments.

Because every step results in a new grammar, production probabilities are fit to observed data by running at most 50 rounds of EM after every step listed above.[2] We focus on our contribution — the coupling step — and direct those interested in details regarding splitting/merging to (Petrov et al., 2006).

Let $\mathcal{T}$ be a treebank and let $\mathcal{F}$ be the set of all possible fragments in $\mathcal{T}$. Define a tree $T \in \mathcal{T}$ as a composition of fragments $\{F_i\}_{i=1}^{n} \subseteq \mathcal{F}$, with $T = F_1 \circ \cdots \circ F_n$. We use X to refer to an arbitrary fragment, with r_X being the root of X. Two fragments X and Y may compose (couple), which we denote by $X \circ Y$.[3] We assume that X and Y may couple only if $X \circ Y$ is an observed subtree.

3.1 Coupling Procedure

While Petrov et al. (2006) posit all refinements simulatenously and then retract half, applying this strategy to the coupling step would result in a combinatorial explosion. We control this combinatorial increase in three ways. First, we assume binary trees. Second, we introduce a constraint set $\mathcal{C} \subseteq \mathcal{F}$ that dictates what fragments are permitted to compose into larger fragments. Third, we adopt the iterative approach of split-merge and incrementally make our grammar more complex by forbidding a fragment from participating in "chained couplings:" $X \circ Y \circ Z$ is not allowed unless either $X \circ Y$ or $Y \circ Z$ is a valid fragment in the previous grammar (and the chained coupling is allowed by $\mathcal{C}$). Note that setting $\mathcal{C} = \emptyset$ results in standard split/merge, while $\mathcal{C} = \mathcal{F}$ results in a latently-refined dop-1 model.

We say that $\langle XY \rangle$ represents a valid coupling of X and Y only if $X \circ Y$ is allowed by $\mathcal{C}$, whereas $\overline{\langle XY \rangle}$ represents an invalid coupling if $X \circ Y$ is not allowed by $\mathcal{C}$. Valid couplings result in new fragments. (We describe how to obtain $\mathcal{C}$ in §3.3.)

Given a constraint set $\mathcal{C}$ and a current grammar $\mathcal{G}$, we construct a new grammar $\mathcal{G}'$. For every fragment $F \in \mathcal{G}$, hypothesize a fragment $F' = F \circ C$, provided $F \circ C$ is allowed by$\mathcal{C}$. In order to add F and F' to $\mathcal{G}'$, we assign an initial probability to both fragments (§3.2), and then use EM to determine appropriate weights. We do not explicitly remove smaller fragments from the grammar, though it is possible for weights to vanish throughout iterations of EM.

Note that a probabilistic TSG fragment may be uniquely represented as its constituent CFG rules: make the root of every internal depth-one subtree unique (have unit probability) and place the entirety of the TSG weight on the root depth-one rule. This representation has multiple benefits: it not only allows TSG induction within the split/merge framework, but it also provides a straight-forward way to use the inside-outside algorithm.

[1] Code available at `cs.jhu.edu/~ferraro`.

[2] We additionally apply Petrov et al. (2006)'s smoothing step between **split** and **merge**.

[3] Technically, the composition operator ($\circ$) is ambiguous if there is more than one occurrence of r_Y in the frontier of X. Although notation augmentations could resolve this, we rely on context for disambiguation.

3.2 Fragment Probability Estimation

First, we define a count function c over fragments by

$$c(X) = \sum_{T \in \mathcal{P}(\mathcal{T})} \sum_{\tau \in T} \delta_{X,\tau}, \quad (1)$$

where $\mathcal{P}(\mathcal{T})$ is a parsed version of $\mathcal{T}$, τ is a subtree of T and $\delta_{X,\tau}$ is 1 iff X matches τ.[4] We may then count fragment co-occurrence by

$$\sum_{Y} c(X \circ Y) = \sum_{Y:\overline{\langle XY \rangle}} c(X \circ Y) + \sum_{Y:\langle XY \rangle} c(X \circ Y).$$

Prior to running inside-outside, we must reallocate the probability mass from the previous fragments to the hypothesized ones. As this is just a temporary initialization, can we allocate mass as done when splitting, where each rule's mass is uniformly distributed, modulo tie-breaking randomness, among its refinement offspring? Split/merge only hypothesizes that a node should have a particular refinement, but by learning subtrees our coupling method hypothesizes that deeper structure may better explain data. This leads to the realization that a symbol may both subsume, and be subsumed by, another symbol in the same coupling step; it is not clear how to apply the above redistribution technique to our situation.

However, even if uniform-redistribution could easily be applied, we would like to be able to indicate how much we "trust" newly hypothesized fragments. We achieve this via a parameter $\gamma \in [0, 1]$: as $\gamma \to 1$, we wish to move more of $\mathbf{P}[X \mid r_X]$ to $\mathbf{P}[\langle XY \rangle \mid r_X]$. Note that we need to know which fragments L couple below with X ($\langle XL \rangle$), and which fragments U couple above ($\langle UX \rangle$).

For reallocation, we remove a fraction of the number of occurrences of top-couplings of X:

$$\hat{c}(X) = 1 - \gamma \frac{\sum_{Y:\langle XY \rangle} c(X \circ Y)}{\sum_{Y} c(X \circ Y)}, \quad (2)$$

and some proportion of the number of occurrences of bottom-couplings of X:

$$\text{sub}(X) = \frac{\sum_{U:\langle UX \rangle} c(U \circ X)}{\sum_{\substack{U,L:\langle UL \rangle \\ r_X = r_L}} c(U \circ L)}. \quad (3)$$

[4] We use a parsed version because there are no labeled internal nodes in the original treebank.

To prevent division-by-zero (e.g., for pre-terminals), (2) returns 1 and (3) returns 0 as necessary.

Given any fragment X in an original grammar, let ρ be its conditional probability: $\rho = \mathbf{P}[X \mid r_X]$. For a new grammar, define the new conditional probability for X to be

$$\mathbf{P}[X \mid r_X] \propto \rho \cdot |\hat{c}(X) - \text{sub}(X)|, \quad (4)$$

and

$$\mathbf{P}[\langle XY \rangle \mid r_X] \propto \gamma \rho \frac{c(X \circ Y)}{\sum_Y c(X \circ Y)} \quad (5)$$

for applicable Y.

Taken together, equations (4) and (5) simply say that X must yield some percentage of its current mass to its hypothesized relatives $\langle XY \rangle$, the amount of which is proportionately determined by $\hat{c}$. But we may also hypothesize $\langle ZX \rangle$, which has the effect of removing (partial) occurrences of X.[5]

Though we would prefer posterior counts of fragments, it is not obvious how to efficiently obtain posterior "bigram" counts of arbitrarily large latent TSG fragments (i.e., $c(X \circ Y)$). We therefore obtain, in linear time, Viterbi counts using the previous best grammar. Although this could lead to count sparsity, in practice our previous grammar provides sufficient counts across fragments.

3.3 Coupling from Common Subtrees

We now turn to the question of how to acquire the constraint set $\mathcal{C}$. Drawing on the discussion in §2.2, the constraint set should, with little effort, enforce sparsity. Similarly to our experiments in classification with TSGs (Ferraro et al., 2012), we extract a list of the K most common subtrees of size at most R, which we refer to as $\mathcal{F}_{\langle R,K \rangle}$. Note that if $F \in \mathcal{F}_{\langle R,K \rangle}$, then all subtrees F' of F must also be in $\mathcal{F}_{\langle R,K \rangle}$.[6] Thus, we may incrementally build $\mathcal{F}_{\langle R,K \rangle}$ in the following manner: given r, for $1 \leq r \leq R$, maintain a ranking S, by frequency, of all fragments of size r; the key point is that S may be built from $\mathcal{F}_{\langle r-1,K \rangle}$. Once all fragments of size r have been considered, retain only the top K fragments of the ranked set $\mathcal{F}_{\langle r,K \rangle} = \mathcal{F}_{\langle r-1,K \rangle} \cup S$.

[5] If $\hat{c}(X) = \text{sub}(X)$, then define Eqn. (4) to be ρ.

[6] Analogously, if an n-gram appears K times, then all constituent m-grams, $m < n$, must also appear at least K times.

This incremental approach is appealing for two reasons: (1) practically, it helps temper the growth of intermediate rankings $\mathcal{F}_{\langle r,K \rangle}$; and (2) it provides two tunable parameters R and K, which relate to the base measure and concentration parameter of previous work (Post and Gildea, 2009; Cohn et al., 2010). We enforce sparsity by thresholding at every iteration.

4 Datasets

We perform a qualitative analysis of fragments learned on datasets for two languages: the Korean Treebank v2.0 (Han and Ryu, 2005) and a comparably-sized portion of the WSJ portion of the Penn Treebank (Marcus et al., 1993). The Korean Treebank (KTB) has predefined splits; to be comparable for our analysis, from the PTB we used §2-3 for training and §22 for validation (we refer to this as wsj2-3). As described in Chung et al. (2010), although Korean presents its own challenges to grammar induction, the KTB yields additional difficulties by including a high occurrence of very flat rules (in 5K sentences, there are 13 NP rules with at least four righthand side NPs) and a coarser nonterminal set than that of the Penn Treebank. On both sets, we run for two iterations.

Recall that our algorithm is designed to induce a state-split TSG on a binarized tree; as neither dataset is binarized in native form we apply a left-branching binarization across all trees in both collections as a preprocessing step. Petrov et al. (2006) found different binarization methods to be inconsequential, and we have yet to observe significant impact of this binarization decision (this will be considered in more detail in future work).

Recently Petrov et al. (2011) provided a set of coarse, "universal" (as measured across 22 languages), part-of-speech tags. We explore here the interaction of this tagset in our model on wsj2-3: call this modified version uwsj2-3, on which we run three iterations. By further coarsening the PTB tags, we can ask questions such as: what is the refinement pattern? Can we identify linguistic phenomena in a different manner than we might without the universal tag set? Then, as an extreme, we replace all POS tags with the same symbol "X," to investigate what predicate/argument relationships can be derived: we call this set xwsj2-3 and run four times on it.[7]

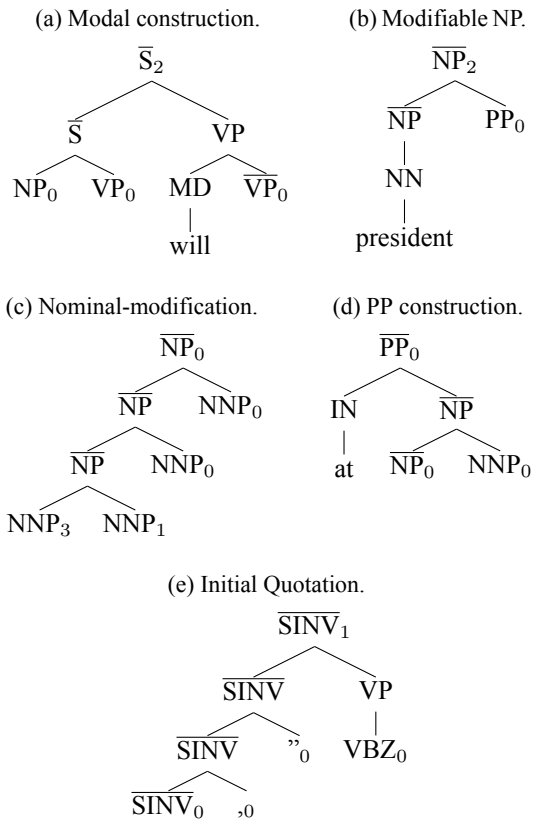


Figure 2: Example fragments learned on wsj2-3.

5 Fragment Analysis

In this section we analyze hand-selected preliminary fragments and lexical clusterings our system learns.

WSJ, §2-3 As Figure 2 illustrates, after two iterations we learn various types of descriptive lexicalized and unlexicalized fragments. For example, Figure 2a concisely creates a four-step modal construction (*will*), while 2b demonstrates how a potentially useful nominal can be formed. Further, learned fragments may generate phrases with multiple nominal modifiers (2c), and lexicalized PPs (2d).

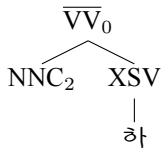
Note that phrases such as $\overline{\text{NP}}_0$ and $\overline{\text{VP}}_0$ are often lexicalized themselves (with determiners, common verbs and other constructions), though omitted due to space constraints; these lexicalized phrases could be very useful for 2a (given the incremental

[7] While the universal tag set has a Korean mapping, the symbols do not coincide with the KTB symbols.

(a) Common noun refinements.

	NNC		
0	경우	이날	현재
	case	*this day*	*at the moment*
1	국제	경제	세계
	international	*economy*	*world*
2	관련	발표	보도
	related	*announcement*	*report*

(b) Verbal inflection.



(c) Adjectival inflection.

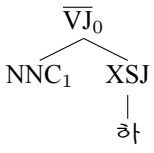


Figure 3: Clusters and fragments for the KTB.

coupling employed, 2a could not have been further expanded in two iterations). Figure 2c demonstrates how TSGs and latent annotations are naturally complementary: the former provides structure while the latter describes lexical distributions of nominals.

Figure 2e illustrates a final example of syntactic structure, as we begin to learn how to properly analyze a complex quotation. A full analysis requires only five TSG rules while an equivalent CFG-only construction requires eight.

KTB2 To illustrate emergent semantic and syntactic patterns, we focus on common noun (NNC) refinements. As seen in Table 3a, top words from NNC_0 represent time expressions and planning-related. As a comparison, two other refinements, NNC_1 and NNC_2, are not temporally representative. This distinction is important as NNC_0 easily yields adverbial phrases, while the resultant adverbial yield for either NNC_1 or NNC_2 is much smaller.

Comparing NNC_1 and NNC_2, we see that the highest-ranked members of the latter, which include *report* and *announcement*, can be verbalized by appending an appropriate suffix. Nouns under NNC_1, such as *economy* and *world*, generally are subject to adjectival, rather than verbal, inflection. Figures 3b and 3c capture these verbal and adjectival inflections, respectively, as lexicalized TSG fragments.

WSJ, §2-3, Universal Tag Set In the preliminary work done here, we find that after a small number of iterations we can identify various cluster classifications for different POS tags. Figures 4a, 4b and 4c provide examples for NOUN, VERB and PRON, respectively. For NOUNs we found that refinements correspond to agentive entities (refinements 0, 1, e.g., corporations or governments), market or stock concepts (2), and numerically-modifiable nouns (7). Some refinements overlapped, or contained common nouns usable in many different contexts (3).

Similarly for VERBs (4b), we find suggested distinctions among action (1) and belief/cognition (2) verbs.[8] Further, some verb clusters are formed of eventive verbs, both general (3) and domain-specific (0). Another cluster is primarily of copula/auxiliary verbs (7). The remaining omitted categories appear to overlap, and only once we examine the contexts in which they occur do we see they are particularly useful for parsing FRAGs.

Though NOUN and VERB clusters can be discerned, there tends to be overlap among refinements that makes the analysis more difficult. On the other hand, refinements for PRON (4c) tend to be fairly clean and it is generally simple to describe each: possessives (1), personified *wh*-words (2) and general *wh*-words (3). Moreover, both subject (5) and object (6) are separately described.

Promisingly, we learn interactions among various refinements in the form of TSG rules, as illustrated by Figures 4d-4g. While all four examples involve VERBs it is enlightening to analyze a VERB's refinement and arguments. For example, the refinements in 4d may lend a simple analysis of financial actions, while 4e may describe different NP interactions (note the different refinement symbols). Different VERB refinements may also coordinate, as in 4f, where participle or gerund may help modify a main verb. Finally, note how in 4g, an object pronoun correctly occurs in object position. These examples suggest that even on coarsened POS tags, our method is able to learn preliminary joint syntactic and lexical relationships.

WSJ, §2-3, Preterminals as X In this experiment, we investigate whether the manual annotations of Petrov et al. (2011) can be re-derived through first reducing one's non-terminal tagset to the symbol X and splitting until finding first the coarse grain

[8] The next highest-ranked verbs for refinement 1 include *received*, *doing* and *announced*.

(a) Noun refinements.

	NOUN		
0	Corp	Big	Co.
1	Mr.	U.S.	New
2	Bush	prices	trading
3	Japan	September	Nissan
7	year	%	months

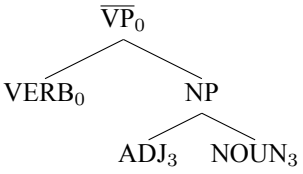
(b) Verb refinements.

	VERB				
0	says	said	sell	buy	rose
1	have	had	has	been	made
2	said	says	say	added	believe
3	sold	based	go	trading	filed
7	is	are	be	was	will

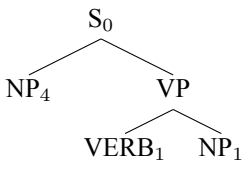
(c) Pronoun refinements.

	PRON		
1	its	his	your
2	who	whom	—
3	what	whose	What
5	it	he	they
6	it	them	him

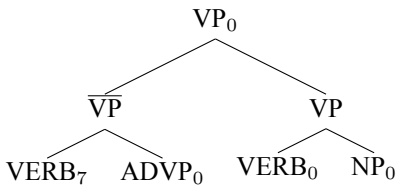
(d) VP structure.



(e) Declarative sentence.



(f) Multiple VP interactions.



(g) Accusative use.

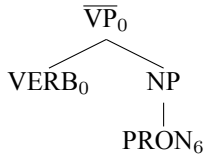


Figure 4: Highest weighted representatives for lexical categories (4a-4c) and learned fragments (4d-4g), for uwsj2-3.

	X			Universal Tag
0	two	market	brain	NOUN
1	's	said	says	VERB
2	%	company	year	NOUN
3	it	he	they	PRON
5	also	now	even	ADV
6	the	a	The	DET
7	10	1	all	NUM
9	.	–	...	.
10	and	or	but	CONJ
12	which	that	who	PRON
13	is	was	are	VERB
14	as	of	in	ADP
15	up	But	billion	ADP

Table 2: Top-three representatives for various refinements of X, with reasonable analogues to Petrov et al. (2011)'s tags. Universal tag recovery is promising.

tags of the universal set, followed by finer-grain tags from the original treebank. Due to the loss of lexical information, we run our system for four iterations rather than three.

As observed in Table 2, there is strong overlap observed between the induced refinements and the original universal tags. Though there are 16 refinements of X, due to lack of cluster coherence not all are listed. Those tags and unlisted refinements seem to be interwoven in a non-trivial way. We also see complex refinements of both open- and closed-class words occurring: refinements 0 and 2 correspond with the open-class NOUN, while refinements 3 and 12, and 14 and 15 both correspond with the closed classes PRON and ADP, respectively. Note that 1 and 13 are beginning to split verbs by auxiliaries.

6 Conclusion

We have shown that TSGs may be encoded and induced within a framework of syntactic latent annotations. Results were provided for induction using the English Penn, and Korean Treebanks, with further experiments based on the Universal Part of Speech tagset. Examples shown suggest the promise of our approach, with future work aimed at exploring larger datasets using more extensive computational resources.

Acknowledgements Thank you to the reviewers for helpful feedback, and to Johns Hopkins HLTCOE for providing support. We would also like to thank Byung Gyu Ahn for graciously helping us analyze the Korean results. Any opinions expressed in this work are those of the authors.

References

Mohit Bansal and Dan Klein. 2010. Simple, accurate parsing with an all-fragments grammar. In *Proceedings of ACL*, pages 1098–1107. Association for Computational Linguistics.

Rens Bod. 1993. Using an annotated corpus as a stochas-

tic grammar. In *Proceedings of EACL*, pages 37–44. Association for Computational Linguistics.

Rens Bod. 2001. What is the minimal set of fragments that achieves maximal parse accuracy? In *Proceedings of ACL*, pages 66–73. Association for Computational Linguistics.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI*, pages 598–603.

Tagyoung Chung, Matt Post, and Daniel Gildea. 2010. Factors affecting the accuracy of korean parsing. In *Proceedings of the NAACL HLT Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL)*, pages 49–57, Los Angeles, California, USA, June.

Trevor Cohn, Sharon Goldwater, and Phil Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 548–556, Stroudsburg, PA, USA. Association for Computational Linguistics.

Trevor Cohn, Phil Blunsom, and Sharon Goldwater. 2010. Inducing tree-substitution grammars. *Journal of Machine Learning Research*, 11:3053–3096, December.

Markus Dreyer and Jason Eisner. 2006. Better informed training of latent syntactic features. In *Proceedings of EMNLP*, pages 317–326, Sydney, Australia, July. Association for Computational Linguistics.

Francis Ferraro, Matt Post, and Benjamin Van Durme. 2012. Judging Grammaticality with Count-Induced Tree Substitution Grammars. In *Proceedings of the Seventh Workshop in Innovated Use of NLP for Building Educational Applications*.

Joshua Goodman. 1996. Efficient algorithms for parsing the dop model. In *Proceedings of EMNLP*, pages 143–152.

Na-Rae Han and Shijong Ryu. 2005. Guidelines for Penn Korean Treebank. Technical report, University of Pennsylvania.

Chung-hye Han, Na-Rae Han, and Eon-Suk Ko. 2001. Bracketing guidelines for penn korean treebank. Technical report, IRCS, University of Pennsylvania.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages: Beyond Words*, volume 3, pages 71–122. Springer.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of ACL*, pages 75–82, Stroudsburg, PA, USA. Association for Computational Linguistics.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of ACL-ICCL*, pages 433–440, Stroudsburg, PA, USA. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. In *ArXiv*, April.

Matt Post and Daniel Gildea. 2009. Bayesian learning of a tree substitution grammar. In *Proceedings of ACL-IJCNLP (short papers)*, pages 45–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andreas Zollmann and Khalil Sima'an. 2005. A consistent and efficient estimator for data-oriented parsing. *Journal of Automata Languages and Combinatorics*, 10(2/3):367.

Willem Zuidema. 2007. Parsimonious data-oriented parsing. In *Proceedings of EMNLP-CoNLL*, pages 551–560.