Improving Statistical Parsing By Linguistic Regularization

Igor Malioutov and Robert C. Berwick
igorm@mit.edu, berwick@csail.mit.edu
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

***Abstract*—Statistically-based parsers for large corpora, in particular the Penn Tree Bank (PTB), typically have not used all the linguistic information encoded in the annotated trees on which they are trained. In particular, they have not in general used information that records the effects of derivations, such as empty categories and the representation of displaced phrases, as is the case with passive, topicalization, and wh-constructions. Here we explore ways to use this information to "unwind" derivations, yielding a regularized underlying syntactic structure that can be used as an additional source of information for more accurate parsing. In effect, we make use of two joint sets of tree structures for parsing: the surface structure and its corresponding underlying structure where arguments have been restored to their canonical positions. We present a pilot experiment on passives in the PTB indicating that through the use of these two syntactic representations we can improve overall parsing performance by exploiting transformational regularities, in this way paring down the search space of possible syntactic analyses.**

I. Introduction

Much progress has been made in using statistically-based parsers trained on corpora such as the Penn Tree Bank (PTB), consisting of 1 million words and 219,205 sentences, which augments context free parses with additional markers to capture internal argument structure and various types of syntactic movement [1]. However, such parsers are often still unable to recover correct verb argument structure. For example, in a passive construction such as that in (1) below:

(1) Mary was kissed by the guy with a telescope on the lips.

the PP "on the lips" will be attached, incorrectly, to the PP "with a telescope". In contrast, the corresponding active form, (2) below, is readily parsed correctly by such parsers, not surprisingly because the Subject NP-PP combination is no longer located near the ambiguous PP attachment point:

(2) The guy with a telescope kissed Mary on the lips

Such examples are not just hypothetical. For instance, Figure 1 shows that sentence #404 of the test section 23 of the PTB, *Measuring cups may soon be replaced by tablespoons in the laundry room* . is parsed incorrectly exactly in this way by two state-of-the-art parsers, the Stanford parser [2] and Bikel's re-implementation of the Collins parser [3]. In both cases, the PP *in the laundry room* is incorrectly attached to the object NP *tablespoons*. Examples such as these suggest that verb argument structure might be more easily recoverable when sentence structure is represented in some canonical format that more transparently encodes grammatical relations such as Subject and Object. In other words, if the arguments of predicates are in a fixed syntactic position in training examples, then we might expect that this regularity would be simpler for a statistically-based system to detect and learn. More generally, it has often been observed that what makes natural languages difficult to parse is that phrases are displaced from their canonical positions, not only in passives, but in topicalization, wh-movement, and many similar constructions. Each of these breaks the transparent link between predicates and arguments.

However, information about predicate-argument links is still generally recoverable in the PTB corpora analyses, in the form of empty-node annotations along with indexing of the displaced phrases. (In this respect, the PTB annotation partly resembles the "logical form" of certain linguistic theories.) Figure 1 shows how the PTB annotates displacements. In this sentence, the Noun Phrase *Measuring cups* is labeled as `NP-SBJ1`, that is, an NP in the Subject position, with index 1. This is the same index as the empty Noun Phrase following *replaced*. The underlying semantic object is given by the label of the NP of the "by" phrase, `NP-LGS`, i.e., *tablespoons*. We could therefore use this information to restore the displaced phrases back to their canonical positions, e.g., the NP *tablespoons* would be put back into the Subject position, and *measuring cups* returned to its canonical Object position following the verb. In general, we will call these kinds of reconstructions back into a canonical predicate-argument form *linguistic regularizations*.

We note that several researchers have previously attempted to improve statistical parsing performance via representational changes to the grammar, in the form of either tree-level transformations, or by incorporating other latent information present in the Penn Treebank [4], [5], [6], [7]. Most of these approaches follow the paradigm proposed in [4], whereby the parser is retrained on a transformed version of the training set and then after evaluation the resulting parses are de-transformed and evaluated against the known gold standard annotations.

The research reported here differs from these in at least two critical respects. First, previous work such as that in [8] has

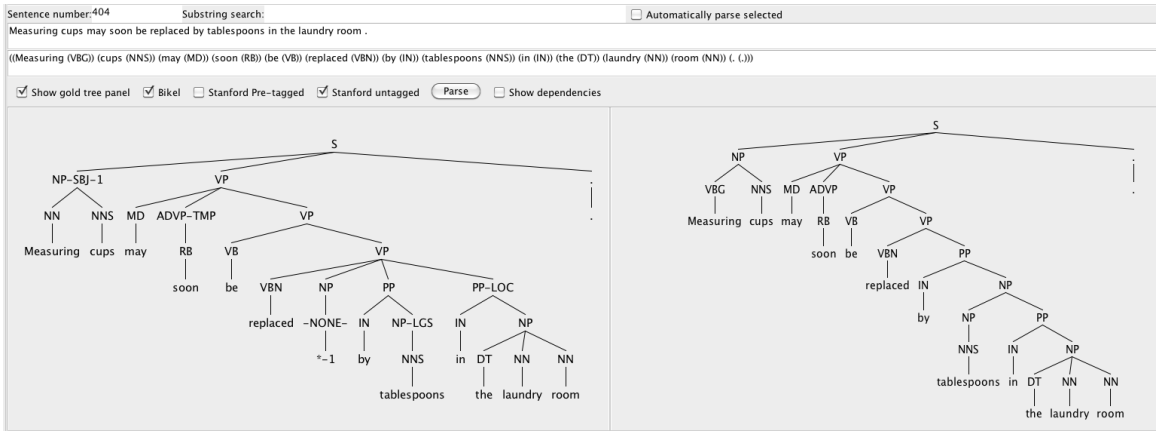

Fig. 1. Stanford and Bikel/Collins mis-parses of sentence number 404 in the PTB. The left-hand panel shows the correct, "gold standard" structure. The right-hand panel displays the result of parsing the same sentence using Bikel's reimplementation of the Collins parser and the Stanford parser, which produce the same erroneous parse.

focused on using additional features in the PTB as a means to improve parsing accuracy, while still others, as in [9] chapter 7, model wh-displacements by means of feature passing. Few approaches have explicitly modeled a separate level of underlying predicate-argument structure. Second, more specifically, the level of syntactic complexity involved in these transformations has been rather limited, and none of the researchers up to the present point have attempted to reassemble the underlying representation of passive constructions.

Following the methodology of [4], we propose to exploit the additional information provided by linguistic regularizations in the following way. First, as suggested above, we can use the annotated PTB training trees to "invert" various displacement operations, returning arguments to their canonical "underlying" positions. In the case of our example sentence, we would derive something like, *Tablespoons may soon replace measuring cups in the laundry room*. We then use the transformed sentences as revised training data for a statistical parser. If the regularization idea is sound, then we would expect improved performance.

II. Passive Transformation: a Pilot Study

We will now show that employing "logical form" structural cues for linguistic regularization can improve parsing performance within the existing Penn Treebank formalism. We selected the passive because it has not, to our knowledge, been tackled in previous work. The experimental setup is as follows. As mentioned, we approach the problem within the framework proposed by [4]. We identify a set of transformations we would like to model in the corpus, transform the input data by performing a set of deterministic 'tree' surgeries on the input parse trees, and then, after re-training, evaluate the resulting parser on a transformed test set.

The first step is to perform tree regular expression (tregex) queries on the corpus to identify the passive constructions in the training data sections of the PTB (sections 02-21). Figure 2 illustrates part of a query for identifying passives in the PTB.

Dataset	Actives	Passives	Total
wsj-02-21	33817	6015 (15.10%)	39832
wsj-23	2052	364 (15.07%)	2416

TABLE I
Penn Treebank Corpus statistics

Second, we must map passive syntactic structures back into their active form counterparts. This mapping is achieved through a sequence of tree-transforms, applied recursively in a bottom-up, right to left fashion using the `Tregex` and `Tsurgeon` toolkit [10]. The following is a simplified version of the sequence of operations required to map the passive form of a sentence to its active counterpart. Note that in some cases, there will be no "by" phrase, that is, no explicit semantic Subject. In these cases, we insert a dummy subject with the part of speech label `TT`, corresponding roughly to *it*.

In all, there are 6,015 passive sentences in the training corpus out of a total of 39,832 sentences. This constitutes 15% of the training data. In the test set, section 23 of the PTB corpus, 364 out of 2416 sentences or 15.1% of the test data can be identified as passives, comparable to the figures observed in the training set (See Table I). The passive construction would therefore seem to provide a good test-bed for a pilot analysis. A 10 percent sample of the identified training set items and all of the test set items were manually checked by a human expert who validated them as true passive constructions.

The third step of the procedure is to re-train and test a statistical parser on the transformed test and training data. We conducted our experiments using Model II of Collins Parser as reimplemented by Bikel [3], and, following the usual methodology, trained on transformed sections 02-21 of the Wall Street Journal PTB (WSJ), and tested the resulting parser on section 23. Additionally, we conducted our experiments on different combinations of transformed and untransformed training and test data, as well as allowing for configurations whereby the test corpora were evaluated on the active and the passive subsets separately. The pilot test results are given in

```
Root /^(.*)-SBJ-(.*)$/=sbj
 > @S|SQ|SINV|S-NOM
    <+(VP|S|SBAR) @VP=hvp
       and
          < /^VB.*$/=bvp
             < /^(am|is|are|was|were|be|'s|being)$/=beverb
          <+(UCP-PRD|ADVP-PRD|S|SBAR|VP) @VP=lvp
             and
                ?< @S|PP=prep
                   ?<+(S|PP) /^(.*)-LGS$/=lgs
                <+(/^VP$/) @VBN=vbn
                   < __=verb
                <+(S|VP|PP-CLR|PP|NP|PP-TMP|S-NOM) /^NP.*$/
                   < /^-NONE-$/=tr
                      < /^(.*)-([0-9]+)$/
```

Fig. 2. Example of a Tregex query identifying simple English passive constructions.

```
move lvp $+ hvp
delete hvp
move lgs $+ sbj
move sbj $+ tr
delete tr
delete prep
excise sbj sbj
relabel lgs /^(.*)-LGS/ #1
```

Fig. 3. Example tree mapping operations for converting passive to active sentences.

table II.

First, we note that the baseline parser (BASE-*) performed markedly better on the active sentence set than on the passive construction subset of the WSJ corpus section 23 (88.27% vs. 87.75% recall). This lower score is to be expected, since the passive construction exhibits longer-range movement and constitutes only 15% of the training data.

On the full test set (2416 trees), the retrained model (TRANS-2) beat the baseline (BASE-1) by 0.12% absolute recall (88.29% vs 88.17%) and 0.11% absolute precision. On the active sentence subset that constitutes about 85% of the test corpus, the model outperforms the baseline by 0.19 percent in recall – a statistically significant difference at the 0.05 level (p-value = 0.029) as computed by a stratified shuffling test with 10,000 iterations. While this may seem like a small performance gain, in the context of a trained parsing system that is known to be operating at close to a theoretical ceiling, this is in fact a real performance increase.

Furthermore, we note that recent work has demonstrated that the eval-b may not be an appropriately granulated metric to measure performance on parse constructions with deep dependencies, which holds true for passives [11].

More concretely, to give an idea of an error that is corrected by regularization, in Figure 4 we display the parser's output of the transformed example sentence, *Tablespoons may soon replace...* The parser outputs a tree that is 100% correct.

To give a broader picture of where the performance improvement comes from, as another example, figure 5 displays an example from section 23 of the PTB, sentence # 722, *According to analysts , profits were also helped by successful cost-cutting measures at Newsweek .*, that is parsed incorrectly in its unregularized form, with a misplaced PP high attachment for *at Newsweek*. This yields a p(recision) score of 91.67% and a r(ecall) score of 84.6% using the standard evalb measure. As the figure shows, after regularization this sentence is now parsed with perfect recall and precision and a correct PP attachment under the NP.

Many other mis-parsed passives from the test dataset are parsed correctly after regularization. In all, out of 364 test sentence passives, 74 improved after regularization. Many of these improvements appear to be due to correction of mis-analyzed PP attachments, as anticipated.

However, the simple regularization carried out in the pilot study can also lead to worse performance; 95 out of 364 test sentence passives were parsed worse than before. It is these cases that reduce the performance gain of regularization in our pilot study. Figures 6 and 7 illustrate one example of this effect. Sentence #2274 in test section 23, the passive sentence, *Tandem 's new high-end computer is called Cyclone .*, is parsed with perfect precision and recall before regularization, though with an arguably incorrect gold-standard bracketing: both an empty Subject NP followed by a predicate NP *Cyclone* are dominated by an S. As Figure 7 shows, after regularization, the re-trained parser mis-analyzes this structure with both the restored Subject NP *Tandem 's* and the predicate NP *Cyclone* combined as a single NP (precision=71.433%, recall=83.33%). It seems likely that examples like these might be successfully analyzed if the gold-standard was assigned a more accurate "small clause" type structure.

Other regularization failures occur where there is no following PP phrase in the original sentence to be mis-parsed, and where the regularization leads to a complex structure with the potential for misanalysis. For instance, the section 23 passive sentence #269, *The land to be purchased by the joint venture has n't yet received zoning and other approvals required for development , and part of Kaufman & Broad 's job will be to obtain such approvals .* requires the NP *the joint venture* to be restored as the Subject of *receive*. However, the re-trained

experiment id	training set	test set	recall	precision	POS	size
BASE-1	wsj-02-21 untrans	wsj-23-full-untrans	88.17	88.36	96.87	2416
BASE-2	wsj-02-21 untrans	wsj-23-full-trans	87.89	88.08	96.73	2416
BASE-3	wsj-02-21 untrans	wsj-23-psv-untrans	87.75	87.96	97.40	364
BASE-4	wsj-02-21 untrans	wsj-23-psv-trans	86.28	86.43	96.65	364
BASE-5	wsj-02-21 untrans	wsj-23-active	88.27	88.45	96.75	2052
TRANS-1	wsj-02-21 trans	wsj-23-full-untrans	88.26	88.48	96.86	2416
TRANS-2	wsj-02-21 trans	wsj-23-full-trans	88.29	88.47	96.82	2416
TRANS-3	wsj-02-21 trans	wsj-23-psv-untrans	87.39	87.65	97.27	364
TRANS-4	wsj-02-21 trans	wsj-23-psv-trans	87.51	87.62	97.02	364
TRANS-5	wsj-02-21 trans	wsj-23-active	88.46	88.66	96.77	2052
SBASE	wsj-02-21 untrans	wsj-23-psv-special	88.12	88.22	97.02	364
STRANS	wsj-02-21 trans	wsj-23-psv-special	89.30	89.38	97.25	364

TABLE II
PARSING RESULTS ON THE ORIGINAL (BASE) AND TRANSFORMED (TRANS) PENN TREEBANK (PTB) DATA

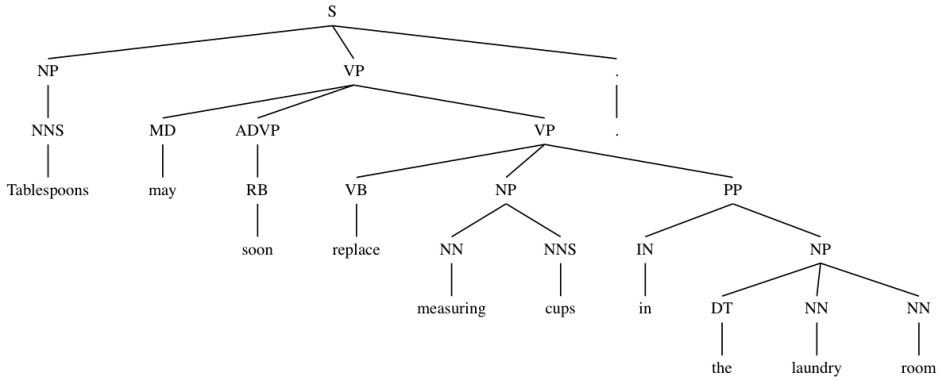

Fig. 4. The Bikel/Collins parser correctly analyzes the "tablespoon" sentence after regularization.

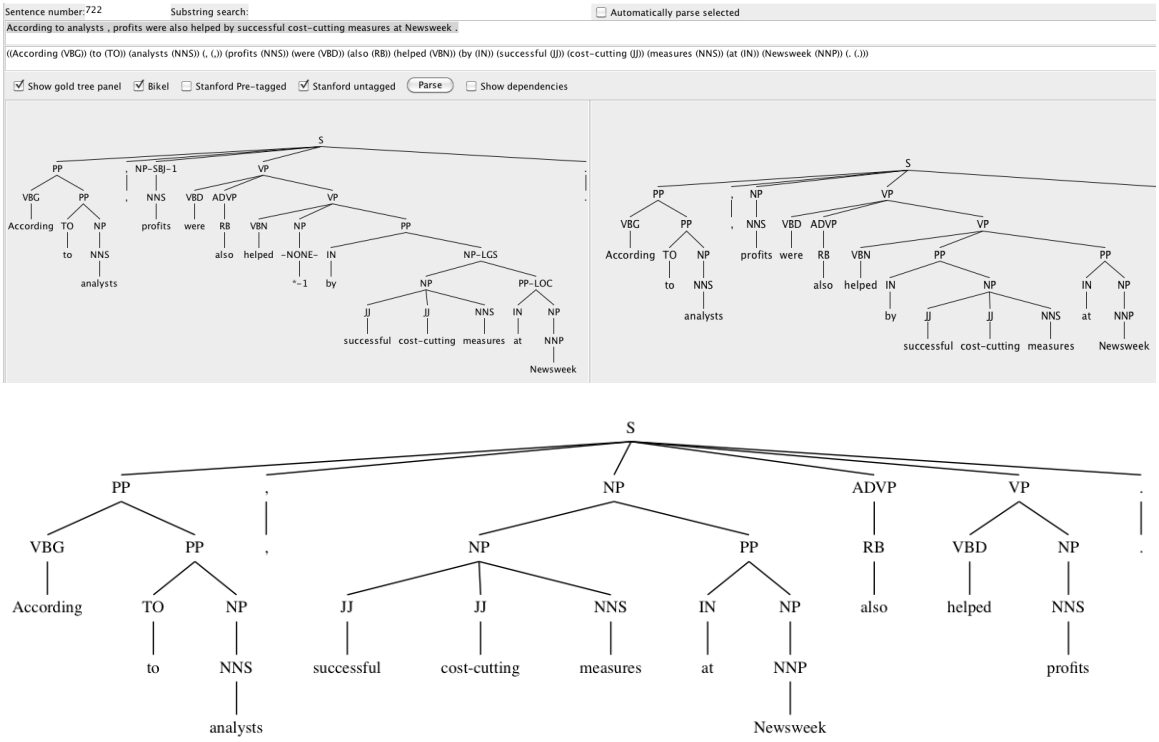

Fig. 5. The Bikel/Collins parser mis-analyzes of sentence # 722 in section 23 of the PTB. The top half of the figure shows the result of parsing the original sentence. The bottom half shows the result of parsing the same sentence correctly after the regularization procedure described in the main text.

parser incorrectly analyzes the regularized sentence. In part this may be the result of not completely reconstructing the underlying form; in this instance, where there is a relative clause *the land purchased by the joint venture*, the object of *receive*, *the land*, is not explicitly restored to its underlying position after the verb. Such complexity has tendency to lead to mis-analysis, and a more complete reconstruction of such relative clauses might repair such instances.

Note that even though on the passive subset (364 trees) the baseline outperforms the transformed model by 0.24% recall, the result is not statistically significant (p-value=0.295). Taken together, the results indicate that retraining significantly improves the performance of the parser on active sentence constructions, while not incurring a statistically significant loss on passives. In fact, the retrained model is much more robust with respect to untransformed passives, only exhibiting a 0.12% loss in precision, whereas the baseline suffers almost a 1.5% degradation (TRANS-3 vs. TRANS-4).

Discussion and Conclusions

The pilot experiment showed that statistically significant improvements in parsing could be achieved by regularizing passive argument structure. However, passive regularization also led to worse performance in some cases. A more careful, case-by-case analysis of these examples would seem warranted, because it appears from a superficial examination of the examples where parsing performance degrades that in each instance the regularization method has partly failed, sometimes introducing additional complex structure. If so, then further improvement may be possible if one can more accurately reconstruct the underlying form, either for small clauses or for relative clauses.

In a final set of validation experiments we examined the effect of selectively unwinding certain passives into their underlying logical form, while leaving others in their original surface form. This is an oracle experiment, whereby we evaluate the parser only on the surface forms that achieve better performance under the retrained parsing model. That is, we assume the presence of an "ominiscient" selection procedure that allows us to decide whether the instance to be parsed for testing first needs to be transformed or whether it is more desirable to leave it in its original form. Note that in practice, we would not have access to such a procedure. However, it is instructive to carry out such an experiment, as it allows us to gauge the best possible (upper bound) performance for the using an "unwound" logical form. This result indicates that we can obtain an upper bound of 89.30% recall, as much as a full percentage point improvement over the baseline by applying the transformations on a selective basis. Further analysis of the results shows that this effect is achieved due to cases where displaced modifiers in the passive construction impact negatively on the parser's attachment decisions. This oracle experiment demonstrates that it is desirable to come up with a general method to determine whether to unwind a parse in the training corpus and hence be able to use surface and deep structure form representations concurrently. So, while our results demonstrate that training a parser on transformed passives improves parsing in general, the oracle experiment also shows that selectively transforming the parses results in even greater gains. The gains demonstrated in figures 4 and 5 are subsumed by a system that uses such a criterion.

In future work, we intend to apply the regularization more broadly to other types of displacements, such as topicalization and dislocation structures. We predict that these will provide additional parsing improvements, possibly approaching the levels achievable only through parse re-ranking.

More generally, we note that the use of paired surface and underlying structures may provide great power not only in improving parsing, but also for providing a means to learn new rules to span the space of grammatical forms that have never been seen in training data, a major roadblock in state-of-the-art statistical systems. This is because our regularization approach bears important parallels to one of the few complete, mathematically established learnability results for a complete grammatical theory, that by Wexler, Hamburger, and Culicover [12]. The Wexler et al. approach is based on a similar idea: the learner is assumed to be able to reconstruct the underlying "D-structure" corresponding to surface sentences, and from this pairing, hypothesize a possible mapping between the two. It remains for future research to determine whether this can be done for other displaced phrases in the PTB more generally.

Finally, we note that in more recent grammatical theories, argument structure is regularized to an even greater degree by means of a vP-VP "shell structure" of branching nodes, that place Subject and then Direct Object and Indirect Object NPs in specific positions [13]. We could readily expand our approach to this notion of regularization, which might provide a statistically-based, machine learning system with additional regularities that are more easily learnable from training data alone.

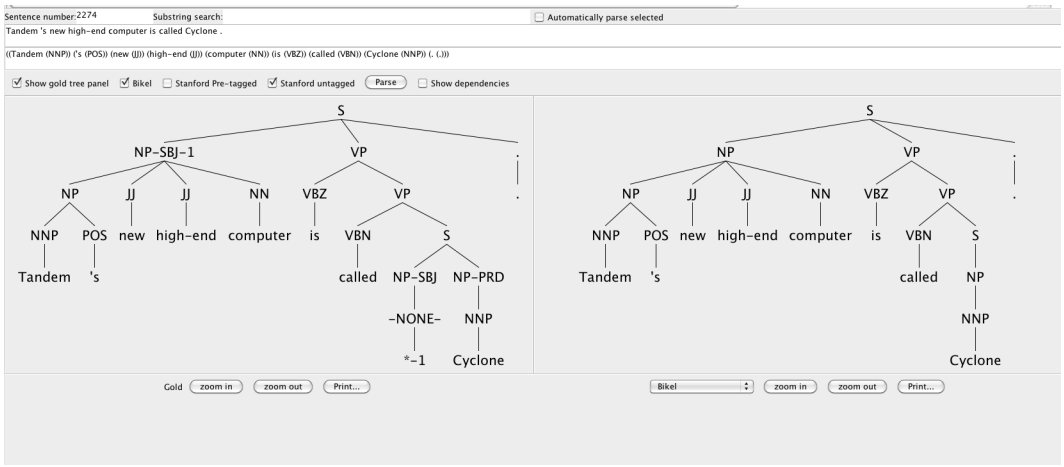

Fig. 6. The Bikel/Collins parser analysis of sentence #2274 of section 23 of the PTB. The gold standard annotation is on the left, the parser output on the right.

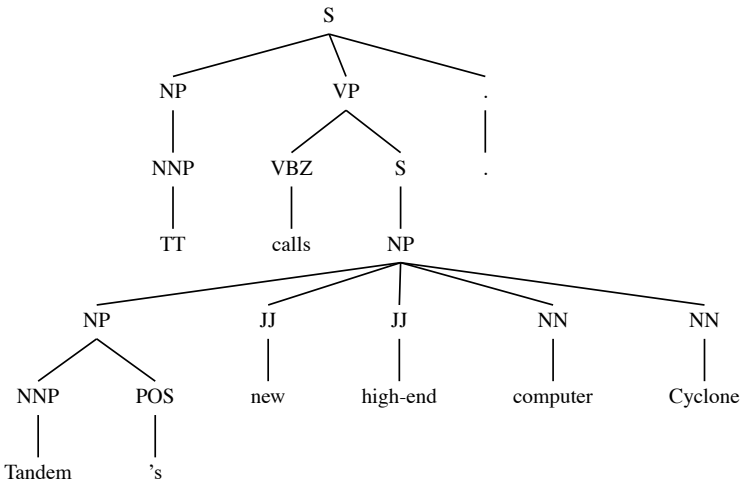

Fig. 7. The parse of regularized sentence #2274 mis-analyzes the NP – NP structure under a single NP, precision=71.433%, recall=83.333%.

REFERENCES

[1] M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The Penn Treebank: Annotating predicate argument structure," *ARPA Human Language Technology Workshop*, pp. 114–119, 1994.

[2] D. Klein and C. Manning, "Fast exact inference with a factored model for natural language parsing," in *Advances in Neural Information Processing Systems*, Cambridge, MA, 2003, pp. 3–10.

[3] D. M. Bikel, "Intricacies of collinsṕarsing model," *Computational Linguistics*, vol. 30, no. 4, pp. 479–511, 2004.

[4] M. Johnson, "Pcfg models of linguistic tree representations," *Computational Linguistics*, vol. 24, no. 4, pp. 613–632, 1998.

[5] J. Eisner, "Smoothing a probabilistic lexicon via syntactic transformations," Ph.D. dissertation, University of Pennsylvania, July 2001.

[6] D. Chiang and D. M. Bikel, "Recovering latent information in treebanks," in *Proceeding of COLING*, 2002.

[7] R. Levy and C. D. Manning, "Deep dependencies from context-free statistical parsers: correcting the surface dependency approximation," in *Proceedings of the ACL*, 2004.

[8] R. Levy, "Probabilistic models of word order and syntactic discontinuity," Ph.D. dissertation, Stanford University, 2006.

[9] M. Collins, "Head-driven statistical models for natural language parsing," Ph.D. dissertation, University of Pennsylvania, 1999.

[10] R. Levy and G. Andrew, "Tregex and tsurgeon: tools for querying and manipulating tree data structures," in *LREC*, 2006.

[11] L. Rimell, S. Clark, and M. Steedman, "Unbounded dependency recovery for parser evaluation." *Proceedings of EMNLP*, pp. 813–821, 2009.

[12] K. Wexler and P. Culicover, *Formal Principles of Language Acquisition*. Cambridge, MA, USA: MIT Press, 1983.

[13] K. Hale and S. Keyser, "On argument structure and the lexical representation of syntactic relations," in *The View from Building 20*, K. Hale and S. Keyser, Eds. Cambridge, MA: MIT Press, 1993, pp. 53–110.