Parsing the Wall Street Journal with the Inside-Outside Algorithm

Yves Schabes Michal Roth Randy Osborne
Mitsubishi Electric Research Laboratories
Cambridge MA 02139
USA
(schabes/roth/osborne@merl.com)

Abstract

We report grammar inference experiments on partially parsed sentences taken from the Wall Street Journal corpus using the inside-outside algorithm for stochastic context-free grammars. The initial grammar for the inference process makes no assumption of the kinds of structures and their distributions. The inferred grammar is evaluated by its predicting power and by comparing the bracketing of held out sentences imposed by the inferred grammar with the partial bracketings of these sentences given in the corpus. Using part-of-speech tags as the only source of lexical information, high bracketing accuracy is achieved even with a small subset of the available training material (1045 sentences): 94.4% for test sentences shorter than 10 words and 90.2% for sentences shorter than 15 words.

1 Introduction

Most broad coverage natural language parsers have been designed by incorporating hand-crafted rules. These rules are also very often further refined by statistical training. Furthermore, it is widely believed that high performance can only be achieved by disambiguating lexically sensitive phenomena such as prepositional attachment ambiguity, coordination or subcategorization.

So far, grammar inference has not been shown to be effective for designing wide coverage parsers.

Baker (1979) describes a training algorithm for stochastic context-free grammars (SCFG) which can be used for grammar reestimation (Fujisaki et al. 1989, Sharman et al. 1990, Black et al. 1992, Briscoe and Waegner 1992) or grammar inference from scratch (Lari and Young 1990). However, the application of SCFGs and the original inside-outside algorithm for grammar inference has been inconclusive for two reasons. First, each iteration of the algorithm on a grammar with n nonterminals requires $O(n^3|w|^3)$ time per training sentence w. Second, the inferred grammar imposes bracketings which do not agree with linguistic judgments of sentence structure.

Pereira and Schabes (1992) extended the inside-outside algorithm for inferring the parameters of a stochastic context-free grammar to take advantage of constituent bracketing information in the training text. Although they report encouraging experiments (90% bracketing accuracy) on language transcriptions in the Texas Instrument subset of the Air Travel Information System (ATIS), the small size of the corpus (770 bracketed sentences containing a total of 7812 words), its linguistic simplicity, and the computation time required to train the grammar were reasons to believe that these results may not scale up to a larger and more diverse corpus.

We report grammar inference experiments with this algorithm from the parsed Wall Street Journal corpus.

The experiments prove the feasibility and effectiveness of the inside-outside algorithm on a large corpus.

Such experiments are made possible by assuming a right branching structure whenever the parsed corpus leaves portions of the parsed tree unspecified. This preprocessing of the corpus makes it fully bracketed. By taking advantage of this fact in the implementation of the inside-outside algorithm, its complexity becomes linear with respect to the input length (as noted by Pereira and Schabes, 1992) and therefore tractable for large corpora.

We report experiments using several kinds of initial grammars and a variety of subsets of the corpus as training data. When the entire Wall Street Journal corpus was used as training material, the time required for training has been further reduced by using a parallel implementation of the inside-outside algorithm.

The inferred grammar is evaluated by measuring the percentage of compatible brackets of the bracketing imposed by the inferred grammar with the partial bracketing of held out sentences. Surprisingly high bracketing accuracy is achieved with only 1042 sentences as training material: 94.4% for test sentences shorter than 10 words and 90.2% for sentences shorter than 15 words. Furthermore, the bracketing accuracy does not drop drastically as longer sentences are considered. These results are surprising since the training uses part-of-speech tags as the only source of lexical information. This raises questions about the statistical distribution of sentence structures observed in naturally occurring text.

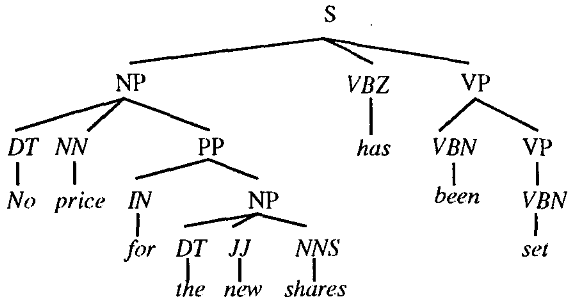
After having described the training material used, we report experiments using several subsets of the available training material and evaluate the effect of the training size on the bracketing performance. Then, we describe a method for reducing the number of parameters in the inferred grammars. Finally, we suggest a stochastic model for inferring labels on the produced binary branching trees.

2 Training Corpus

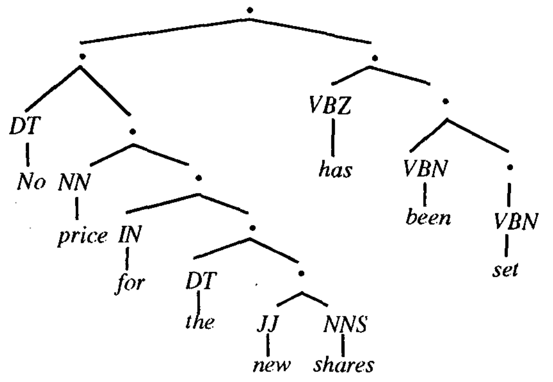
The experiments use texts from the Wall Street Journal Corpus and its partially bracketed version provided by the Penn Treebank (Brill et al., 1990). Out of 38 600 bracketed sentences (914 000 words), we extracted 34500 sentences (817 000 words) as possible source of training material and 4100 sentences (97 000 words) as source for testing. We experimented with several subsets (350, 1095, 8000 and 34500 sentences) of the available training material.

For practical purposes, the part of the tree bank used for training is preprocessed before being used. First, flat portions of parse trees found in the tree bank are turned into a right linear binary branching structure. This enables us to take full advantage of the fact that the extended inside-outside algorithm (as described in Pereira and Schabes, 1992) behaves in linear time when the text is fully bracketed. Then, the syntactic labels are ignored. This allows the reestimation algorithm to distribute its own set of labels based on their actual distribution. We later suggest a method for recovering these labels.

The following is an example of a partially parsed sentence found in the Penn Treebank:



The above parse corresponds to the fully bracketed unlabeled parse



found in the training corpus. The experiments reported in this paper use only the part-of-speech sequences of this corpus and the resulting fully bracketed parses. For the above example, the following bracketing is used in the training material:

(DT (NN (IN (DT (JJ NNS)))) (VBZ (VBN VBN)))

3 Inferring Bracketings

For the set of experiments described in this section, the initial grammar consists of all 4095 possible Chomsky Normal Form rules over 15 nonterminals (X_i, $1 < i < 15$) and 48 terminal symbols (t_m, $1 < m < 48$) for part-of-speech tags (the same set as the one used in the Penn Treebank):

$$X_i \Rightarrow X_j X_k$$

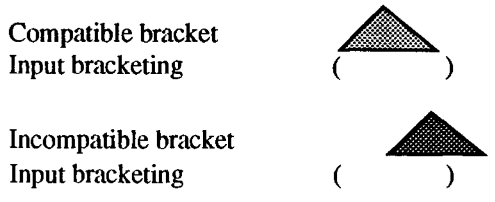
$$X_i \Rightarrow t_m$$

The parameters of the initial stochastic context-free grammar are set randomly while maintaining the proper conditions for stochastic context-free grammars.[1]

Using the algorithm described in Pereira and Schabes (1992), the current rule probabilities and the parsed training set C are used to estimate the expected frequencies of each rule. Once these frequencies are computed over each bracketed sentence c in the training set, new rule probabilities are assigned in a way that increases the estimated probability of the bracketed training set. This process is iterated until the increase in the estimated probability of the bracketed training text becomes negligible, or equivalently, until the decrease in cross entropy (negative log probability)

$$\hat{H}(C, G) = -\frac{\sum_{c \in C} \log P(c)}{\sum_{c \in C} |c|}$$

becomes negligible. In the above formula, the probability P(c) of the partially bracketed sentence c is computed as the sum of the probabilities of all derivations compatible with the bracketing of the sentence. This notion of compatible bracketing is defined in details in Pereira and Schabes (1992). Informally speaking, a derivation is compatible with the bracketing of the input given in the tree bank, if no bracket imposed by the derivation crosses a bracket in the input.



As training material, we selected randomly out of the available training material 1042 sentences of length shorter than 15 words. For evaluation purposes, we also randomly selected 84 sentences of length shorter than 15 words among the test sentences.

Figure 1 shows the cross entropy of the training after each iteration. It also shows for each iteration the cross entropies $\hat{H}$ of 84 sentences randomly selected among the test sentences of length shorter than 15 words. The cross entropy decreases as more iterations are performed and no over training is observed..

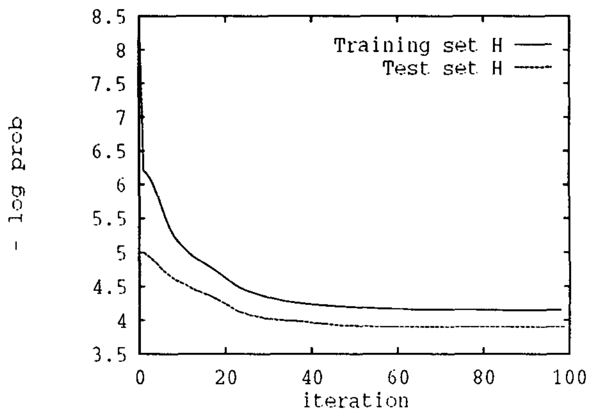


Figure 1. Training and Test Set -log prob

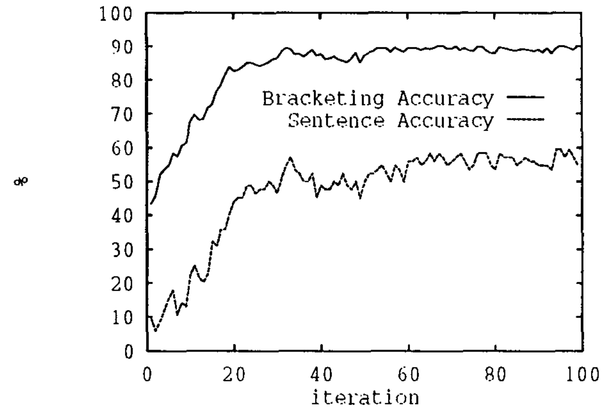


Figure 2. Bracketing and sentence accuracy of 84 test sentences shorter than 15 words.

To evaluate the quality of the analyses yielded by the inferred grammars obtained after each iteration, we used a Viterbi-style parser to find the most likely analyses of sentences in several test samples, and compared them with the Treebank partial bracketings of the sentences of those samples. For each sample, we counted the percent-

[1] The sum of the probabilities of the rules with same left hand side must be one.

age of brackets of the most likely analysis that are not "crossing" the partial bracketing of the same sentences found in the Treebank. This percentage is called the *bracketing accuracy* (see Pereira and Schabes, 1992 for the precise definition of this measure). We also computed the percentage of sentences in each sample in which no crossing bracket was found. This percentage is called the *sentence accuracy.*

Figure 2 shows the bracketing and sentence accuracy for the same 84 test sentences.

Table 1 shows the bracketing and sentence accuracy for test sentences within various length ranges. High bracketing accuracy is obtained even on relatively long sentences. However, as expected, the sentence accuracy decreases rapidly as the sentences get longer.

Length	0-10	0-15	10-19	20-30
Bracketing Accuracy	94.4%	90.2%	82.5%	71.5%
Sentence Accuracy	82%	57.1%	30%	6.8%

TABLE 1. Bracketing Accuracy on test sentences of different lengths (using 1042 sentences of lengths shorter than 15 words as training material).

Table 2 compares our results with the bracketing accuracy of analyses obtained by a systematic right linear branching structure for all words except for the final punctuation mark (which we attached high).[2] We also evaluated the stochastic context-free grammar obtained by collecting each level of the trees found in the training tree bank (see Table 2).

Length .	**0-10**	**0-15**	**10-19**	**20-30**
Inferred grammar	94.4%	90.2%	82.5%	71.5%
Right linear trees	76%	70%	63%	50%
Treebank Grammar	46%	31%	25%	

TABLE 2. Bracketing accuracy of the inferred grammar, of right linear structures and of the Treebank grammar.

Right linear structures perform surprisingly well. Our results improve by 20 percentage points upon this base line performance. These results suggest that the distribution of sentence structure in naturally occurring text is simpler than one may have thought, especially since only part-of-speech tags were used. This may suggest the existence of clusters of trees in the training material. However, using the number of crossing brackets as a distance between trees, we have been unable to reveal the existence of clusters.

The grammar obtained by collecting rules from the tree bank performs very poorly. One can conclude that the labels used in the tree bank do not have any statistical property. The task of inferring a stochastic grammar from a tree bank is not trivial and therefore requires statistical training.

In the appendix we give examples of the most likely analyses output by the inferred grammar on several test sentences

In Table 3, different subsets of the available training sentences of lengths up to 15 words long and the grammars were evaluated on the same set of test sentences of lengths shorter than 15 words. The size of the training set does not seem to affect the performance of the parser.

Training Size (sentences)	350	1095	8000
Bracketing Accuracy	89.37%	90.22%	89.86%
Sentence Accuracy	52.38%	57.14%	55.95%

TABLE 3. Effect of the size of the training set on the bracketing and sentence accuracy.

However if one includes all available sentences (34700 sentences), for the same test set, the bracketing accuracy drops to 84% and the sentence accuracy to 40%.

We have also experimented with the following initial grammar which defines a large number of rules (110640):

$$X_i \Rightarrow X_j X_k$$

$$X_i \Rightarrow t_i$$

In this grammar, each non-terminal symbol is uniquely associated with a terminal symbol. We observed overtraining with this grammar and better statistical convergence was obtained, however the performance of the parser did not improve.

[2] We thank Eric Brill and David Yarowsky for suggesting these experiments.

4 Reducing the Grammar Size and Smoothing Issues

As grammars are being inferred at each iteration, the training algorithm was designed to guarantee that no parameter was set below some small threshold. This constraint is important for smoothing. It implies that no rule ever disappears at a reestimation step.

However, once the final grammar is found, for practical purposes, one can reduce the number of parameters being used. For example, the size of the grammar can be reduced by eliminating the rules whose probabilities are below some threshold or by keeping for each non-terminal only the top rules rewriting it.

However, one runs into the risk of not being able to parse sentences given as input. We used the following smoothing heuristics.

Lexical rule smoothing. In the case no rule in the grammar introduces a terminal symbol found in the input string, we assigned a lexical rule ($X_i \Rightarrow t_m$) with very low probability for all non-terminal symbols. This case will not happen if the training is representative of the lexical items.

Syntactic rule smoothing. When the sentence is not recognized from the starting symbol, we considered all possible non-terminal symbols as starting symbols and considered as starting symbol the one that yields the most likely analysis. Although this procedure may not guarantee that all sentences will be recognized, we found it is very useful in practice.

When none of the above procedures enable parsing of the sentence, we used the entire set of parameters of the inferred grammar (this was never the case on the test sentences we considered).

For example, the grammar whose performance is depicted in Table 2 defines 4095 parameters. However, the same performance is achieved on these test sets by using only 450 rules (the top 20 binary branching rules $X_i \Rightarrow X_j X_k$for each non-terminal symbol and the top 10 lexical rules $X_i \Rightarrow t_m$ for each non-terminal symbol),

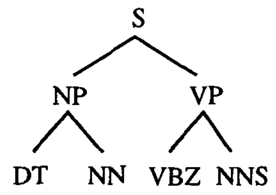
5. Implementation

Pereira and Schabes (1992) note that the training algorithm behaves in linear time (with respect to the sentence length) when the training material consists of fully bracketed sentences. By taking advantage of this fact, the experiments using a small number of initial rules and a small subset of the available training materials do not require a lot of computation time and can be performed on a single workstation. However, the experiments using larger initial grammars or using more material require more computation.

The training algorithm can be parallelized by dividing the training corpus into fixed size blocks of sentences and by having multiple workstations processing each one of them independently. When all blocks have been computed, the counts are merged and the parameters are reestimated. For this purpose, we used PVM (Beguelin et al., 1991) as a mechanism for message passing across workstations.

6. Stochastic Model of Labeling for Binary Branching Trees

The stochastic grammars inferred by the training procedures produce unlabeled parse trees. We are currently evaluating the following stochastic model for labeling a binary branching tree. In this approach, we make the simplifying assumption that the label of a node only depends on the labels of its children. Under this assumption, the probability of labeling a tree is the product of the probability of labeling each level in the tree. For example, the probability of the following labeling:



is $P(S \Rightarrow NP\ VP)\ P(NP \Rightarrow DT\ NN)\ P(VP \Rightarrow VBZ\ NNS)$

These probabilities can be estimated in a simple manner given a tree bank. For example, the probability of labeling a level as $NP \Rightarrow DT\ NN$ is estimated as the number of occurrences (in the tree bank) of $NP \Rightarrow DT\ NN$ divided by the number of occurrences of $X \Rightarrow DT\ NN$ where X ranges over every label.

Then the probability of a labeling can be computed bottom-up from leaves to root. Using dynamic programming on increasingly large subtrees, the labeling with the highest probability can be computed.

We are currently evaluating the effectiveness of this method.

7. Conclusion

The experiments described in this paper prove the effectiveness of the inside-outside algorithm on a large corpus, and also shed some light on the distribution of sentence structures found in natural languages.

We reported grammar inference experiments using the inside-outside algorithm on the parsed Wall Street Journal corpus. The experiments were made possible by turning the partially parsed training corpus into a fully bracketed corpus.

Considering the fact that part-of-speech tags were the only source of lexical information actually used, surprisingly high bracketing accuracy is achieved (90.2% on sentences of length up to 15). We believe that even higher results can be achieved by using a richer set of part-of-speech tags. These results show that the use of simple distributions of constituency structures can provide high accuracy performance for broad coverage natural language parsers.

Acknowledgments

We thank Eric Brill, Aravind Joshi, Mark Liberman, Mitchel Marcus, Fernando Pereira, Stuart Shieber and David Yarowsky for valuable discussions.

References

Baker, J.K. 1979. Trainable grammars for speech recognition. In Jared J. Wolf and Dennis H. Klatt, editors, Speech communication papers presented at the 97^{th} *Meeting of the Acoustical Society of America*, MIT, Cambridge, MA, June.

Adam Beguelin, Jack Dongarra, Al Geist, Robert Manchek, Vaidy Sunderam. July 1991."A Users' guide to PVM Parallel Virtual Machine", Oak Ridge National Lab, TM-11826.

E. Black, S. Abney, D. Flickenger, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. *DARPA Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, California. Morgan Kaufmann.

Ezra Black, John Lafferty, and Salim Roukos. 1992. Development and Evaluation of a Broad-Coverage Probabilistic Grammar of English-Language Computer Manuals. In 20^{th} *Meeting of the Association for Computational Linguistics (ACL' 92)*, Newark, Delaware.

Eric Brill, David Magerman, Mitchell Marcus, and Beatrice Santorini. 1990. Deducing linguistic structure from the statistics of large corpora. In *DARPA Speech and Natural Language Workshop*. Morgan Kaufmann, Hidden Valley, Pennsylvania, June.

Ted Briscoe and Nick Waegner. July 1992. Robust Stochastic Parsing Using the Inside-Outside Algorithm. In *AAAI workshop on Statistically-based Techniques in Natural Language Processing.*

T. Fujisaki, F. Jelinek, J. Cocke, E. Black, and T. Nishino. 1989. A probabilistic parsing method for sentence disambiguation. *Proceedings of the International Workshop on Parsing Technologies*, Pittsburgh, August.

K. Lari and S.J. Young. 1990. The estimation of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech and Language*, 4:35-56.

Pereira, Fernando and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In 20^{th} *Meeting of the Association for Computational Linguistics (ACL' 92)*, Newark, Delaware.

Appendix Examples of parses

The following parsed sentences are the most likely analyses output by the grammar inferred from 1042 training sentences (at iteration 68) for some randomly selected sentences of length not exceeding 10 words. Each parse is preceded by the bracketing given in the Treebank. Sentences output by the parser are printed in bold face and crossing brackets are marked with an asterisk (*).

(((The/DT Celtona/NP operations/NNS) would/MD (become/VB (part/NN (of/IN (those/DT ventures/NNS))))) ./.)
(((The/DT (Celtona/NP operations/NNS)) (would/MD (become/VB (part/NN (of/IN (those/DT ventures/NNS))))))) ./.)

((But/CC then/RB they/PP (wake/VBP up/IN (to/TO (a/DT nightmare/NN)))) ./.)
((But/CC (then/RB (they/PP (wake/VBP (up/IN (to/TO (a/DT nightmare/NN))))))) ./.)

(((Mr./NP Strieber/NP) (knows/VBZ (a/DT lot/NN (about/IN aliens/NNS)))) ./.)
(((Mr./NP Strieber/NP) (knows/VBZ ((a/DT lot/NN) (about/IN aliens/NNS)))) ./.)

(((The/DT companies/NNS) (are/VBP (automotive-emissions-testing/JJ concerns/NNS))) ./.)
(((The/DT companies/NNS) (are/VBP (automotive-emissions-testing/JJ concerns/NNS))) ./.)

(((Chief/JJ executives/NNS and/CC presidents/NNS) had/VBD (come/VBN and/CC gone/VBN) ./.))
(((Chief/JJ (executives/NNS (and/CC presidents/NNS))) (had/VBD (come/VBN (and/CC gone/VBN)))) ./.)

(((How/WRB quickly/RB) (things/NNS change/VBP) ./.))
((How/WRB (* quickly/RB (things/NNS change/VBP) *)) ./.)

((This/DT (means/VBZ ((the/DT returns/NNS) can/MD (vary/VB (a/DT great/JJ deal/NN))))) ./.)
((This/DT (means/VBZ ((the/DT returns/NNS) (can/MD (vary/VB (a/DT (great/JJ deal/NN))))))) ./.)

(((Flight/NN Attendants/NNS) (Lag/NN (Before/IN (Jets/NNS Even/RB Land/VBP)))))
((* Flight/NN (* Attendants/NNS (* Lag/NN (* Before/IN Jets/NNS *) *) *) *) (Even/RB Land/VBP))

((They/PP (talked/VBD (of/IN (the/DT home/NN run/NN)))) ./.)
((They/PP (talked/VBD (of/IN (the/DT (home/NN run/NN))))) ./.)

(((The/DT entire/JJ division/NN) (employs/VBZ (about/IN 850/CD workers/NNS))) ./.)
(((The/DT (entire/JJ division/NN)) (employs/VBZ (about/IN (850/CD workers/NNS)))) ./.)

(((At/IN least/JJS) (before/IN (8/CD p.m/RB)) ./.))
(((At/IN least/JJS) (before/IN (8/CD p.m/RB))) ./.)

((Pretend/VB (Nothing/NN Happened/VBD)))
((* Pretend/VB Nothing/NN *) Happened/VBD)

(((The/DT highlight/NN) :/: (a/DT ``/`` fragrance/NN control/NN system/NN ./. ''/")))
((* (The/DT highlight/NN) (* :/: (a/DT ((``/`` fragrance/NN) (control/NN system/NN))) *) *) (./. ''/"))

(((Stock/NP prices/NNS) (slipped/VBD lower/JJR (in/IN (moderate/JJ trading/NN))) ./.))
(((Stock/NP prices/NNS) (slipped/VBD (lower/JJR (in/IN (moderate/JJ trading/NN))))) ./.)

(((Some/DT jewelers/NNS) (have/VBP (Geiger/NP counters/NNS) (to/TO (measure/VB (topaz/NN radiation/NN)))) ./.))
(((Some/DT jewelers/NNS) (have/VBP ((Geiger/NP counters/NNS) (to/TO (measure/VB (topaz/NN radiation/NN)))))) ./.)

((That/DT ('s/VBZ ((the/DT only/JJ question/NN) (we/PP (need/VBP (to/TO address/VB)))))) ./.)
((That/DT ('s/VBZ ((the/DT (only/JJ question/NN)) (we/PP (need/VBP (to/TO address/VB)))))) ./.)

((She/PP (was/VBD (as/RB (cool/JJ (as/IN (a/DT cucumber/NN)))))) ./.)
((She/PP (was/VBD (as/RB (cool/JJ (as/IN (a/DT cucumber/NN)))))) ./.)

(((The/DT index/NN) (gained/VBD (99.14/CD points/NNS) Monday/NP)) ./.)
(((The/DT index/NN) (gained/VBD ((99.14/CD points/NNS) Monday/NP))) ./.)