Man* vs. Machine: A Case Study in Base Noun Phrase Learning

Eric Brill and Grace Ngai
Department of Computer Science
The Johns Hopkins University
Baltimore, MD 21218, USA
Email: {brill,gyn}@cs.jhu.edu

Abstract

A great deal of work has been done demonstrating the ability of machine learning algorithms to automatically extract linguistic knowledge from annotated corpora. Very little work has gone into quantifying the difference in ability at this task between a person and a machine. This paper is a first step in that direction.

1 Introduction

Machine learning has been very successful at solving many problems in the field of natural language processing. It has been amply demonstrated that a wide assortment of machine learning algorithms are quite effective at extracting linguistic information from manually annotated corpora.

Among the machine learning algorithms studied, rule based systems have proven effective on many natural language processing tasks, including part-of-speech tagging (Brill, 1995; Ramshaw and Marcus, 1994), spelling correction (Mangu and Brill, 1997), word-sense disambiguation (Gale et al., 1992), message understanding (Day et al., 1997), discourse tagging (Samuel et al., 1998), accent restoration (Yarowsky, 1994), prepositional-phrase attachment (Brill and Resnik, 1994) and base noun phrase identification (Ramshaw and Marcus, In Press; Cardie and Pierce, 1998; Veenstra, 1998; Argamon et al., 1998). Many of these rule based systems learn a short list of simple rules (typically on the order of 50-300) which are easily understood by humans.

Since these rule-based systems achieve good performance while learning a small list of simple rules, it raises the question of whether people could also derive an effective rule list manually from an annotated corpus. In this paper we explore how quickly and effectively relatively untrained people can extract linguistic generalities from a corpus as compared to a machine. There are a number of reasons for doing this. We would like to understand the relative strengths and weaknesses of humans versus machines in hopes of marrying their complementary strengths to create even more accurate systems. Also, since people can use their metaknowledge to generalize from a small number of examples, it is possible that a person could derive effective linguistic knowledge from a much smaller training corpus than that needed by a machine. A person could also potentially learn more powerful representations than a machine, thereby achieving higher accuracy.

In this paper we describe experiments we performed to ascertain how well humans, given an annotated training set, can generate rules for base noun phrase chunking. Much previous work has been done on this problem and many different methods have been used: Church's PARTS (1988) program uses a Markov model; Bourigault (1992) uses heuristics along with a grammar; Voutilainen's NPTool (1993) uses a lexicon combined with a constraint grammar; Juteson and Katz (1995) use repeated phrases; Veenstra (1998), Argamon, Dagan & Krymolowski(1998) and Daelemans, van den Bosch & Zavrel (1999) use memory-based systems; Ramshaw & Marcus (In Press) and Cardie & Pierce (1998) use rule-based systems.

2 Learning Base Noun Phrases by Machine

We used the base noun phrase system of Ramshaw and Marcus (R&M) as the machine learning system with which to compare the hu-

*and Woman.

man learners. It is difficult to compare different machine learning approaches to base NP annotation, since different definitions of base NP are used in many of the papers, but the R&M system is the best of those that have been tested on the Penn Treebank.[1]

To train their system, R&M used a 200k-word chunk of the Penn Treebank Parsed Wall Street Journal (Marcus et al., 1993) tagged using a transformation-based tagger (Brill, 1995) and extracted base noun phrases from its parses by selecting noun phrases that contained no nested noun phrases and further processing the data with some heuristics (like treating the possessive marker as the first word of a new base noun phrase) to flatten the recursive structure of the parse. They cast the problem as a transformation-based tagging problem, where each word is to be labelled with a chunk structure tag from the set {I, O, B}, where words marked "I" are inside some base NP chunk, those marked "O" are not part of any base NP, and those marked "B" denote the first word of a base NP which immediately succeeds another base NP. The training corpus is first run through a part-of-speech tagger. Then, as a baseline annotation, each word is labelled with the most common chunk structure tag for its part-of-speech tag.

After the baseline is achieved, transformation rules fitting a set of rule templates are then learned to improve the "tagging accuracy" of the training set. These templates take into consideration the word, part-of-speech tag and chunk structure tag of the current word and all words within a window of 3 to either side of it. Applying a rule to a word changes the chunk structure tag of a word and in effect alters the boundaries of the base NP chunks in the sentence.

An example of a rule learned by the R&M system is: *change a chunk structure tag of a word from I to B if the word is a determiner, the next word is a noun, and the two previous words both have chunk structure tags of I.* In other words, a determiner in this context is likely to begin a noun phrase. The R&M system learns a total of 500 rules.

3 Manual Rule Acquisition

R&M framed the base NP annotation problem as a word tagging problem. We chose instead to use regular expressions on words and part of speech tags to characterize the NPs, as well as the context surrounding the NPs, because this is both a more powerful representational language and more intuitive to a person. A person can more easily consider potential phrases as a sequence of words and tags, rather than looking at each individual word and deciding whether it is part of a phrase or not. The rule actions we allow are:[2]

Add	Add a base NP (bracket a sequence of words as a base NP)
Kill	Delete a base NP (remove a pair of parentheses)
Transform	Transform a base NP (move one or both parentheses to extend/contract a base NP)
Merge	Merge two base NPs

As an example, we consider an actual rule from our experiments:

> Bracket all sequences of words of: one determiner (DT), zero or more adjectives (JJ, JJR, JJS), and one or more nouns (NN, NNP, NNS, NNPS), if they are followed by a verb (VB, VBD, VBG, VBN, VBP, VBZ).

In our language, the rule is written thus:[3]

```
A
(* .)
({1} t=DT) (* t=JJ[RS]?) (+ t=NNP?S?)
({1} t=VB[DGNPZ]?)
```

The first line denotes the action, in this case, Add a bracketing. The second line defines the context preceding the sequence we want to have bracketed — in this case, we do not care what this sequence is. The third line defines the sequence which we want bracketed, and the last

[1] We would like to thank Lance Ramshaw for providing us with the base-NP-annotated training and test corpora that were used in the R&M system, as well as the rules learned by this system.

[2] The rule types we have chosen are similar to those used by Vilain and Day (1996) in transformation-based parsing, but are more powerful.

[3] A full description of the rule language can be found at http://nlp.cs.jhu.edu/~baseNP/manual.

line defines the context following the bracketed sequence.

Internally, the software then translates this rule into the more unwieldy Perl regular expression:

```
s{(([^\s_]+__DT\s+)([^\s_]+__JJ[RS]\s+)*
([^\s_]+__NNP?S?\s+)+)([^\s_]+__VB[DGNPZ]\s+)}
{ ( $1 ) $5 }g
```

The base NP annotation system created by the humans is essentially a transformation-based system with hand-written rules. The user manually creates an ordered list of rules. A rule list can be edited by adding a rule at any position, deleting a rule, or modifying a rule. The user begins with an empty rule list. Rules are derived by studying the training corpus and NPs that the rules have not yet bracketed, as well as NPs that the rules have incorrectly bracketed. Whenever the rule list is edited, the efficacy of the changes can be checked by running the new rule list on the training set and seeing how the modified rule list compares to the unmodified list. Based on this feedback, the user decides whether to accept or reject the changes that were made. One nice property of transformation-based learning is that in appending a rule to the end of a rule list, the user need not be concerned about how that rule may interact with other rules on the list. This is much easier than writing a CFG, for instance, where rules interact in a way that may not be readily apparent to a human rule writer.

To make it easy for people to study the training set, word sequences are presented in one of four colors indicating that they:

1. are not part of an NP either in the truth or in the output of the person's rule set
2. consist of an NP both in the truth and in the output of the person's rule set (i.e. they constitute a base NP that the person's rules correctly annotated)
3. consist of an NP in the truth but not in the output of the person's rule set (i.e. they constitute a recall error)
4. consist of an NP in the output of the person's rule set but not in the truth (i.e. they constitute a precision error)

The actual system is located at `http://nlp.cs.jhu.edu/~basenp/chunking`. A screenshot of this system is shown in figure 4. The correct base NPs are enclosed in parentheses and those annotated by the human's rules in brackets.

4 Experimental Set-Up and Results

The experiment of writing rule lists for base NP annotation was assigned as a homework set to a group of 11 undergraduate and graduate students in an introductory natural language processing course.[4]

The corpus that the students were given from which to derive and validate rules is a 25k word subset of the R&M training set, approximately $\frac{1}{8}$ the size of the full R&M training set. The reason we used a downsized training set was that we believed humans could generalize better from less data, and we thought that it might be possible to meet or surpass R&M's results with a much smaller training set.

Figure 1 shows the final precision, recall, F-measure and precision+recall numbers on the training and test corpora for the students. There was very little difference in performance on the training set compared to the test set. This indicates that people, unlike machines, seem immune to overtraining. The time the students spent on the problem ranged from less than 3 hours to almost 10 hours, with an average of about 6 hours. While it was certainly the case that the students with the worst results spent the least amount of time on the problem, it was not true that those with the best results spent the most time — indeed, the average amount of time spent by the top three students was a little less than the overall average — slightly over 5 hours. On average, people achieved 90% of their final performance after half of the total time they spent in rule writing.

The number of rules in the final rule lists also varied, from as few as 16 rules to as many as 61 rules, with an average of 35.6 rules. Again, the average number for the top three subjects was a little under the average for everybody: 30.3 rules.

[4] These 11 students were a subset of the entire class. Students were given an option of participating in this experiment or doing a much more challenging final project. Thus, as a population, they tended to be the less motivated students.

	TRAINING SET (25K Words)				TEST SET			
	Precision	Recall	F-Measure	$\frac{P+R}{2}$	Precision	Recall	F-Measure	$\frac{P+R}{2}$
Student 1	87.8%	88.6%	88.2	88.2	88.0%	88.8%	88.4	88.4
Student 2	88.1%	88.2%	88.2	88.2	88.2%	87.9%	88.0	88.1
Student 3	88.6%	87.6%	88.1	88.2	88.3%	87.8%	88.0	88.1
Student 4	88.0%	87.2%	87.6	87.6	86.9%	85.9%	86.4	86.4
Student 5	86.2%	86.8%	86.5	86.5	85.8%	85.8%	85.8	85.8
Student 6	86.0%	87.1%	86.6	86.6	85.8%	87.1%	86.4	86.5
Student 7	84.9%	86.7%	85.8	85.8	85.3%	87.3%	86.3	86.3
Student 8	83.6%	86.0%	84.8	84.8	83.1%	85.7%	84.4	84.4
Student 9	83.9%	85.0%	84.4	84.5	83.5%	84.8%	84.1	84.2
Student 10	82.8%	84.5%	83.6	83.7	83.3%	84.4%	83.8	83.8
Student 11	84.8%	78.8%	81.7	81.8	84.0%	77.4%	80.6	80.7

Figure 1: P/R results of test subjects on training and test corpora

In the beginning, we believed that the students would be able to match or better the R&M system's results, which are shown in figure 2. It can be seen that when the same training corpus is used, the best students do achieve performances which are close to the R&M system's — on average, the top 3 students' performances come within 0.5% precision and 1.1% recall of the machine's. In the following section, we will examine the output of both the manual and automatic systems for differences.

5 Analysis

Before we started the analysis of the test set, we hypothesized that the manually derived systems would have more difficulty with potential rules that are effective, but fix only a very small number of mistakes in the training set.

The distribution of noun phrase types, identified by their part of speech sequence, roughly obeys Zipf's Law (Zipf, 1935): there is a large tail of noun phrase types that occur very infrequently in the corpus. Assuming there is not a rule that can generalize across a large number of these low-frequency noun phrases, the only way noun phrases in the tail of the distribution can be learned is by learning low-count rules: in other words, rules that will only positively affect a small number of instances in the training corpus.

Van der Dosch and Daelemans (1998) show that not ignoring the low count instances is often crucial to performance in machine learning systems for natural language. Do the human-written rules suffer from failing to learn these infrequent phrases?

To explore the hypothesis that a primary difference between the accuracy of human and machine is the machine's ability to capture the low frequency noun phrases, we observed how the accuracy of noun phrase annotation of both human and machine derived rules is affected by the frequency of occurrence of the noun phrases in the training corpus. We reduced each base NP in the test set to its POS tag sequence as assigned by the POS tagger. For each POS tag sequence, we then counted the number of times it appeared in the training set and the recall achieved on the test set.

The plot of the test set recall vs. the number of appearances in the training set of each tag sequence for the machine and the mean of the top 3 students is shown in figure 3. For instance, for base NPs in the test set with tag sequences that appeared 5 times in the training corpus, the students achieved an average recall of 63.6% while the machine achieved a recall of 83.5%. For base NPs with tag sequences that appear less than 6 times in the training set, the machine outperforms the students by a recall of 62.8% vs. 54.8%. However, for the rest of the base NPs — those that appear 6 or more times — the performances of the machine and students are almost identical: 93.7% for the machine vs. 93.5% for the 3 students, a difference that is not statistically significant.

The recall graph clearly shows that for the top 3 students, performance is comparable to the machine's on all but the low frequency constituents. This can be explained by the human's

Training set size(words)	Precision	Recall	F-Measure	$\frac{P+R}{2}$
25k	88.7%	89.3%	89.0	89.0
200k	91.8%	92.3%	92.0	92.1

Figure 2: P/R results of the R&M system on test corpus

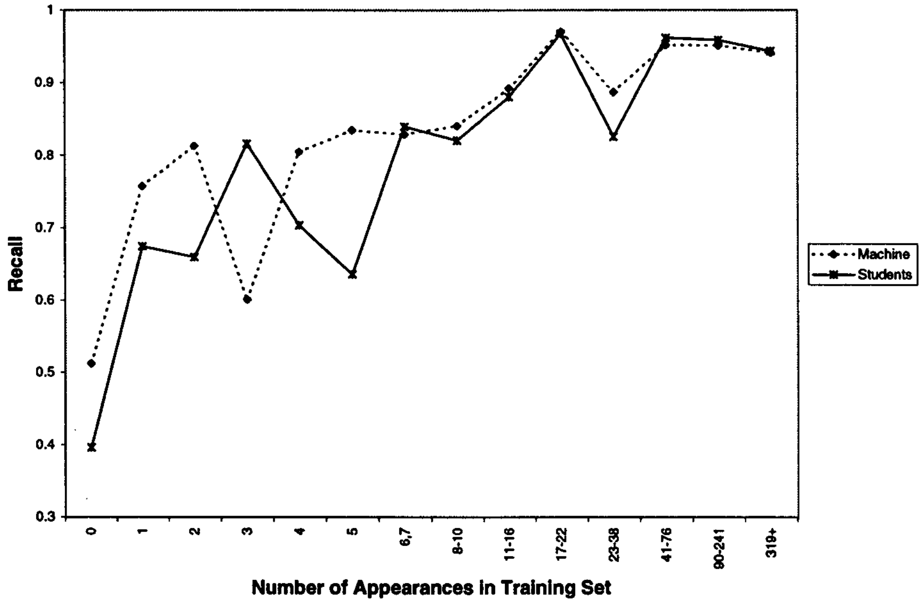


Figure 3: Test Set Recall vs. Frequency of Appearances in Training Set.

reluctance or inability to write a rule that will only capture a small number of new base NPs in the training set. Whereas a machine can easily learn a few hundred rules, each of which makes a very small improvement to accuracy, this is a tedious task for a person, and a task which apparently none of our human subjects was willing or able to take on.

There is one anomalous point in figure 3. For base NPs with POS tag sequences that appear 3 times in the training set, there is a large decrease in recall for the machine, but a large increase in recall for the students. When we looked at the POS tag sequences in question and their corresponding base NPs, we found that this was caused by one single POS tag sequence — that of two successive numbers (CD). The test set happened to include many sentences containing sequences of the type:

...(CD CD) TO (CD CD)...

as in:

(International/NNP Paper/NNP) fell/VBD (1/CD $\frac{3}{8}$/CD) to/TO (51/CD $\frac{1}{2}$/CD)...

while the training set had none. The machine ended up bracketing the entire sequence

1/CD $\frac{3}{8}$/CD to/TO 51/CD $\frac{1}{2}$/CD

as a base NP. None of the students, however, made this mistake.

6 Conclusions and Future Work

In this paper we have described research we undertook in an attempt to ascertain how people can perform compared to a machine at learning linguistic information from an annotated corpus, and more importantly to begin to explore the differences in learning behavior between human and machine. Although people did not match the performance of the machine-learned annotator, it is interesting that these "language novices", with almost no training, were able to come fairly close, learning a small number of powerful rules in a short amount of time on a small training set. This challenges the claim that machine learning offers portability advantages over manual rule writing, seeing that relatively unmotivated people can near-match the best machine performance on this task in so little time at a labor cost of approximately US$40.

We plan to take this work in a number of directions. First, we will further explore whether people can meet or beat the machine's accuracy at this task. We have identified one major weakness of human rule writers: capturing information about low frequency events. It is possible that by providing the person with sufficiently powerful corpus analysis tools to aide in rule writing, we could overcome this problem.

We ran all of our human experiments on a fixed training corpus size. It would be interesting to compare how human performance varies as a function of training corpus size with how machine performance varies.

There are many ways to combine human corpus-based knowledge extraction with machine learning. One possibility would be to combine the human and machine outputs. Another would be to have the human start with the output of the machine and then learn rules to correct the machine's mistakes. We could also have a hybrid system where the person writes rules with the help of machine learning. For instance, the machine could propose a set of rules and the person could choose the best one. We hope that by further studying both human and machine knowledge acquisition from corpora, we can devise learning strategies that successfully combine the two approaches, and by doing so, further improve our ability to extract useful linguistic information from online resources.

Acknowledgements

The authors would like to thank Ryan Brown, Mike Harmon, John Henderson and David Yarowsky for their valuable feedback regarding this work. This work was partly funded by NSF grant IRI-9502312.

References

S. Argamon, I. Dagan, and Y. Krymolowski. 1998. A memory-based approach to learning shallow language patterns. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 67–73. COLING-ACL.

D. Bourigault. 1992. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 30th Annual Meeting of the Association of Computational Linguistics*, pages 977–981. Association of Computational Linguistics.

E. Brill and P. Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the fifteenth International Conference on Computational Linguistics (COLING-1994)*.

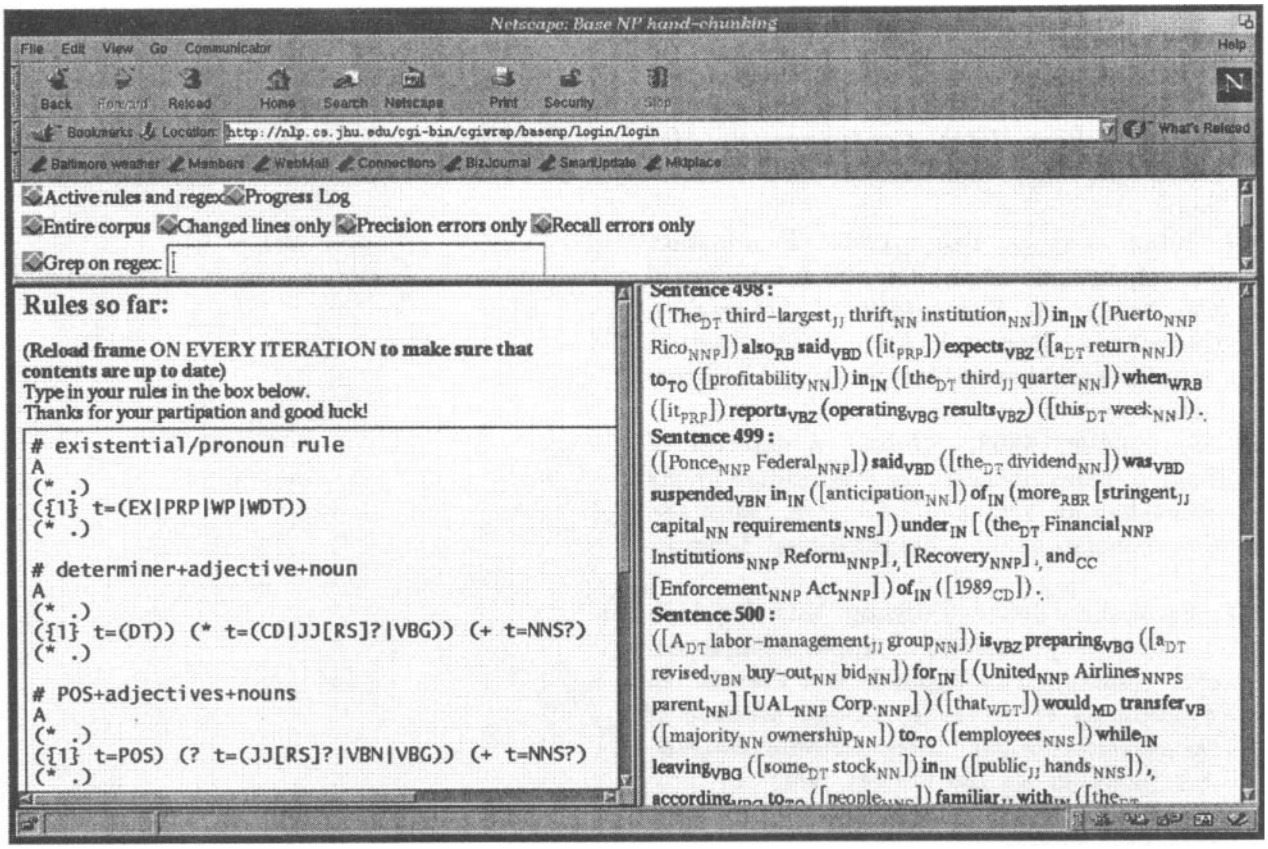
E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, December.

C. Cardie and D. Pierce. 1998. Error-driven pruning of treebank gramars for base noun phrase identification. In *Proceedings of the 36th Annual Meeting of the Association of Computational Linguistics*, pages 218–224. Association of Computational Linguistics.

K. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143. Association of Computational Linguistics.

W. Daelemans, A. van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. In *Machine Learning, special issue on natural language learning*, volume 11, pages 11–43. to appear.

D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain. 1997. Mixed-initiative development of language processing systems. In *Fifth Conference on Applied Natural Language*


Netscape: Base NP hand-chunking
File Edit View Go Communicator
Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location:
What's Related
Baltimore weather Members WebMail Connections BizJournal SmartUpdate Mktplace
Active rules and regex Progress Log
Entire corpus Changed lines only Precision errors only Recall errors only
Grep on regex:

http://nlp.cs.jhu.edu/cgi-bin/cgiwrap/basenp/login/login

Rules so far:

(Reload frame ON EVERY ITERATION **to make sure that contents are up to date)**
Type in your rules in the box below.
Thanks for your partipation and good luck!

```
# existential/pronoun rule
A
(* .)
({1} t=(EX|PRP|WP|WDT))
(* .)

# determiner+adjective+noun
A
(* .)
({1} t=(DT)) (* t=(CD|JJ[RS]?|VBG)) (+ t=NNS?)
(* .)

# POS+adjectives+nouns
A
(* .)
({1} t=POS) (? t=(JJ[RS]?|VBN|VBG)) (+ t=NNS?)
(* .)
```

Sentence 498 :
([The$_{DT}$ third-largest$_{JJ}$ thrift$_{NN}$ institution$_{NN}$]) **in$_{IN}$** ([Puerto$_{NNP}$ Rico$_{NNP}$]) **also$_{RB}$ said$_{VBD}$** ([it$_{PRP}$]) **expects$_{VBZ}$** ([a$_{DT}$ return$_{NN}$]) **to$_{TO}$** ([profitability$_{NN}$]) **in$_{IN}$** ([the$_{DT}$ third$_{JJ}$ quarter$_{NN}$]) **when$_{WRB}$** ([it$_{PRP}$]) **reports$_{VBZ}$** (**operating$_{VBG}$ results$_{VBZ}$**) ([this$_{DT}$ week$_{NN}$]) .$_{.}$
Sentence 499 :
([Ponce$_{NNP}$ Federal$_{NNP}$]) **said$_{VBD}$** ([the$_{DT}$ dividend$_{NN}$]) **was$_{VBD}$ suspended$_{VBN}$ in$_{IN}$** ([anticipation$_{NN}$]) **of$_{IN}$** (**more$_{RBR}$** [**stringent$_{JJ}$** capital$_{NN}$ **requirements$_{NNS}$**]) **under$_{IN}$** [(**the$_{DT}$** Financial$_{NNP}$ Institutions$_{NNP}$ Reform$_{NNP}$] ,$_{,}$ [Recovery$_{NNP}$] ,$_{,}$ and$_{CC}$ [Enforcement$_{NNP}$ Act$_{NNP}$]) **of$_{IN}$** ([1989$_{CD}$]) .$_{.}$
Sentence 500 :
([A$_{DT}$ labor-management$_{JJ}$ group$_{NN}$]) **is$_{VBZ}$ preparing$_{VBG}$** ([a$_{DT}$ revised$_{VBN}$ buy-out$_{NN}$ bid$_{NN}$]) **for$_{IN}$** [(United$_{NNP}$ Airlines$_{NNPS}$ parent$_{NN}$] [UAL$_{NNP}$ Corp.$_{NNP}$]) ([that$_{WDT}$]) **would$_{MD}$ transfer$_{VB}$** ([majority$_{NN}$ ownership$_{NN}$]) **to$_{TO}$** ([employees$_{NNS}$]) **while$_{IN}$ leaving$_{VBG}$** ([some$_{DT}$ stock$_{NN}$]) **in$_{IN}$** ([public$_{JJ}$ hands$_{NNS}$]) ,$_{,}$ **according$_{VBG}$ to$_{TO}$** ([people$_{NNS}$]) **familiar$_{JJ}$ with$_{IN}$** ([the$_{DT}$

Figure 4: Screenshot of base NP chunking system

Processing, pages 348–355. Association for Computational Linguistics, March.

W. Gale, K. Church, and D. Yarowsky. 1992. One sense per discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*, pages 233–237.

J. Juteson and S. Katz. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27.

L. Mangu and E. Brill. 1997. Automatic rule acquisition for spelling correction. In *Proceedings of the Fourteenth International Conference on Machine Learning*, Nashville, Tennessee.

M. Marcus, M. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

L. Ramshaw and M. Marcus. 1994. Exploring the statistical derivation of transformational rule sequences for part-of-speech tagging. In *The Balancing Act: Proceedings of the ACL Workshop on Combining Symbolic and Statistical Approaches to Language*, New Mexico State University, July.

L. Ramshaw and M. Marcus. In Press. Text chunking using transformation-based learning. In *Natural Language Processing Using Very large Corpora*. Kluwer.

K. Samuel, S. Carberry, and K. Vijay-Shanker. 1998. Dialogue act tagging with transformation-based learning. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, volume 2. Association of Computational Linguistics.

A. van der Dosch and W. Daelemans. 1998. Do not forget: Full memory in memory-based learning of word pronunciation. In *New Methods in Language Processing*, pages 195–204. Computational Natural Language Learning.

J. Veenstra. 1998. Fast NP chunking using memory-based learning techniques. In *BENELEARN-98: Proceedings of the Eighth Belgian-Dutch Conference on Machine Learning*, Wageningen, the Netherlands.

M. Vilain and D. Day. 1996. Finite-state parsing by rule sequences. In *International Conference on Computational Linguistics*, Copenhagen, Denmark, August. The International Committee on Computational Linguistics.

A Voutilainen. 1993. *NPTool*, a detector of English noun phrases. In *Proceedings of the Workshop on Very Large Corpora*, pages 48–57. Association for Computational Linguistics.

D. Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 88–95, Las Cruces, NM.

G. Zipf. 1935. *The Psycho-Biology of Language.* Houghton Mifflin.