

# A Machine Learning Approach to Modeling Scope Preferences\*

Derrick Higgins  
University of Chicago

Jerrold Sadock  
University of Chicago

*This paper describes a corpus-based investigation of quantifier scope preferences. Following recent work on multi-modular grammar frameworks in theoretical linguistics, and a long history of combining multiple information sources in natural language processing, we treat scope as a distinct module of grammar from the syntax. This module incorporates multiple sources of evidence regarding the most likely scope reading for a sentence, and is entirely data-driven. This paper reports the results of using tools from machine learning practice in designing a quantifier-scope module, and training on data from the Penn Treebank. We wish to focus attention on the issue of scope prediction, which has been largely ignored in theoretical linguistics, and to explore different models of the interaction between syntax and quantifier scope.*

## 1 Overview

This paper addresses the issue of determining the most accessible quantifier scope reading for a sentence. *Quantifiers* are elements of natural and logical languages (such as *each*, *no*, and *some* in English, and  $\forall$  and  $\exists$  in predicate calculus) which have certain semantic properties. Loosely speaking, they express that a proposition holds for some proportion of a set of individuals. One peculiarity of these expressions is that there can be semantic differences which depend on the order in which the quantifiers are interpreted. These are known as *scope* differences.

- (1) Everyone likes two songs on this album.

As an example of the sort of interpretive differences we are talking about, consider the sentence in (1). There are two readings of this sentence, which depend on which of the

---

\* The authors are grateful for an Academic technology Innovation Grant from the University of Chicago, which helped to make this work possible, and to John Goldsmith, Terry Regier, and Anne Pycha, whose advice and collaboration have considerably aided this research. Any remaining errors are, of course, our own.

two quantified expressions *everyone* and *two songs on this album* takes wide scope. The first reading, in which *everyone* takes wide scope, simply expresses that every person has a certain preference, not necessarily related to anyone else's. This reading can be paraphrased as: "pick any person, that person will like two songs on this album." The second reading, in which *everyone* takes narrow scope, implies that there are two specific songs on the album which everyone is fond of, say *Blue Moon* and *My Way*.

In theoretical linguistics, attention has been primarily focussed on the issue of *scope generation*. Researchers applying the techniques of Quantifier Raising and Cooper Storage have been concerned mainly with enumerating all of the scope readings for a sentence which are possible, without regard to their relative likelihood or naturalness. Recently, however, linguists such as Kuno, Takami, and Wu (1999) have begun to turn their attention to *scope prediction*, or determining the relative accessibility of different scope readings.

In computational linguistics, more attention has been paid to the factors which determine scope preferences. Systems such as the SRI Core Language Engine (Moran, 1988; Moran and Pereira, 1992), LUNAR (Woods, 1986), and TEAM (Martin, Appelt, and Pereira, 1986) have employed *scope critics* which use heuristics to decide between alternative scopings. However, the rules which these systems use in making quantifier scope decisions are motivated only by the researchers' intuitions, and no empirical results have been published regarding their accuracy.

In this paper, we use the tools of machine learning to construct a data-driven model of quantifier scope preferences. For theoretical linguistics, this model serves as an illustration that Kuno, Takami, and Wu's approach can capture some of the clearest generalizations about quantifier scoping. For computational linguistics, this study provides a baseline result on the task of scope prediction, with which other scope critics can be compared. In addition, it is the first empirical investigation we are aware of which collects data of

any kind regarding the relative frequency of different quantifier scope readings in English text.<sup>1</sup>

Section 2 briefly discusses treatments of scoping issues in theoretical linguistics, while Section 3 reviews the computational work which has been done on natural language quantifier scope. In Section 4 we introduce the models which we use to predict quantifier scoping, as well as the data on which they are trained and tested. Section 5 combines the scope model of the previous section with a PCFG model of syntax, and addresses the issue of whether these two modules of grammar ought to be combined in serial, with information from the syntax feeding the quantifier scope module, or in parallel, with each module constraining the structures provided by the other.

## 2 Approaches to Quantifier Scope in Theoretical Linguistics

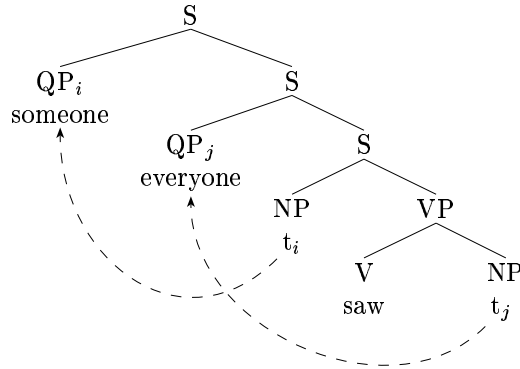
Most, if not all, linguistic treatments of quantifier scope have closely integrated it with the way in which the syntactic structure of a sentence is built up. Montague (1973) used a syntactic rule to introduce a quantified expression into a derivation at the point where it was to take scope, while Generative Semantic analyses such as McCawley (1998) represented the scope of quantification at Deep Structure, transformationally lowering quantifiers into their surface positions during the course of the derivation. More recent work in the interpretive paradigm takes the opposite approach, extracting quantifiers from their surface positions to their scope positions by means of a Quantifier Raising transformation (May, 1985; Aoun and Li, 1993; Hornstein, 1995). Another popular technique is to percolate scope information up through the syntactic tree using Cooper Storage (Cooper, 1983; Hobbs and Shieber, 1987; Pollard, 1989; Nerbonne, 1993; Park, 1995; Pollard and Yoo, 1998).

---

<sup>1</sup> However, see Carden (1976) for a questionnaire-based approach to gathering data on the accessibility of different quantifier scope readings.

**Figure 1**

Simple illustration of the Quantifier Raising approach to quantifier scope generation.



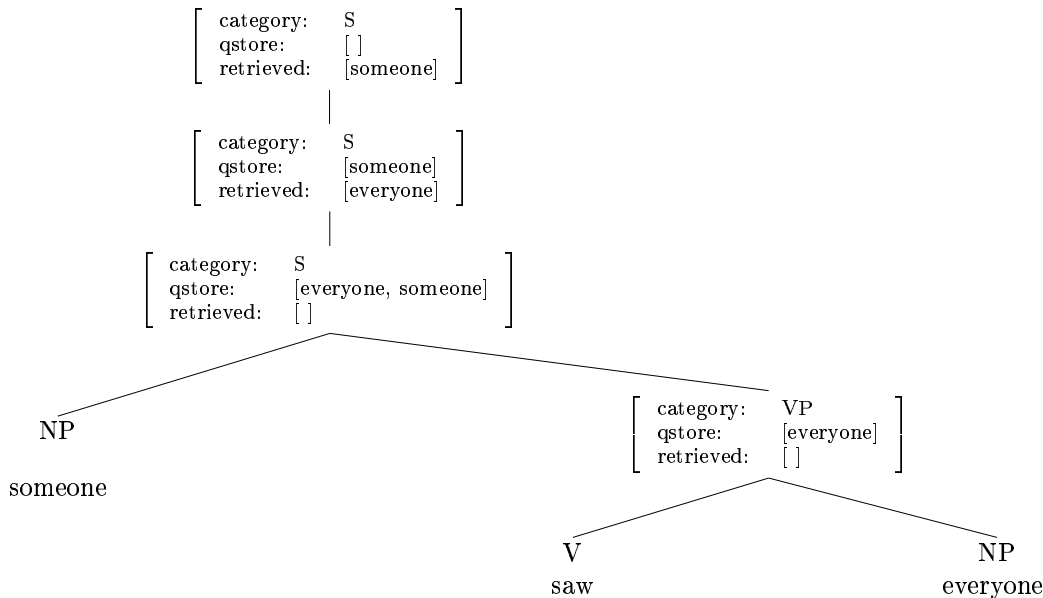
The Quantifier Raising (QR) approach to dealing with scope in linguistics consists in the claim that there is a covert transformation applying to syntactic structures, which moves quantified elements out of the position in which they are found on the surface, and raises them to a higher position which reflects their scope. The various incarnations of this strategy differ in the precise characterization of this Quantifier Raising transformation, what conditions are placed upon it, and what tree-configurational relationship is required for one operator to take scope over another. The general idea of QR is represented in Figure 1, a schematic analysis of the reading of the sentence *Someone saw everyone* in which *someone* takes wide scope (i.e., ‘there is some person  $x$  such that for all persons  $y$ ,  $x$  saw  $y$ ’).

In the Cooper Storage approach, quantifiers are gathered into a *store* and passed upward through a syntactic tree. At certain nodes along the way, quantifiers may be *retrieved* from the store and take scope. The relative scope of quantifiers is determined by where each quantifier is retrieved from the store, with higher quantifiers taking wide scope over lower ones. As with QR, different authors implement this scheme in slightly different ways, but the simplest case is represented in Figure 2, the Cooper Storage analog of Figure 1.

These structural approaches, QR and Cooper storage, have in common that they only

**Figure 2**

Simple illustration of the Cooper Storage approach to quantifier scope generation.



allow syntactic factors to have any effect on the scope readings which are available for a given sentence. They are also similar in only addressing the issue of scope generation, or identifying all and only the accessible readings for each sentence. That is to say, they do not address the issue of the relative salience of these readings.

Kuno, Takami, & Wu (KT&W) (1999; 2001) propose to model the scope of quantified elements with a set of interacting expert systems, which basically consists of a weighted vote taken of the various factors which may influence scope readings. This model is meant not only to account for scope generation, but also “the relative strengths of the potential scope interpretations of a given sentence” (1999, p. 63). They illustrate the plausibility of this approach in their paper by presenting a number of examples which are accounted for fairly well even when allowing an unweighted vote of the factors to be taken.

So, for example, in KT&W’s (49b), repeated here as (2), the correct prediction is made, that the sentence is unambiguous with the first quantified NP taking wide scope over the second (the reading in which we don’t all have to hate the same people). Table

**Table 1**

Voting to determine optimal scope readings for quantifiers, according to KT&W (1999).

	<i>many of us/you</i>	<i>some of them</i>
Baseline:	✓	✓
Subject Q:	✓	
Lefthand Q:	✓	
Speaker/Hearer Q:	✓	
<b>Total:</b>	<b>4</b>	<b>1</b>

1 illustrates how the votes of each of KT&W’s “experts” contribute to this outcome. Since the expression *many of us/you* receives more votes, and the numbers for the two competing quantified expressions are quite far apart, the first one is predicted to take wide scope unambiguously.

(2) Many of us/you hate some of them.

Some adherents of the structural approaches also seem to acknowledge the necessity of eventually coming to terms with the factors which play a role in determining scope preferences in language. Aoun and Li (2000) claim that the lexical scope preferences of quantifiers “are not ruled out under a structural account” (p. 140). It is clear from the surrounding discussion, though, that they intend such lexical requirements to be taken care of in some non-syntactic component of grammar. While Kuno, Takami, & Wu’s dialogue with Aoun & Li in *Language* has been portrayed by both sides as a debate over the correct way of modeling quantifier scope, they are not really modeling the same things. While Aoun and Li (1993) provide an account of scope generation, KT&W (1999) intend to model both scope generation and scope prediction. The model of scope preferences provided in this paper is an empirically-based refinement of the approach taken by KT&W, but in principle it is consistent with a structural account of scope generation.

### 3 Approaches to Quantifier Scope in Computational Linguistics

Many studies have dealt with the issue of scope generation from a computational perspective, such as Pereira (1990) and Park (1995). Attempts have also been made in computational work to extend a pure Cooper Storage approach to handle scope prediction. Hobbs and Shieber (1987) discuss the possibility of incorporating some sort of ordering heuristics into the SRI scope generation system, in the hopes of producing a ranked list of possible scope readings, but ultimately are forced to admit that “[t]he modifications turn out to be quite complicated if we wish to order quantifiers according to lexical heuristics, such as having ‘each’ out-scope ‘some’. Because of the recursive nature of the algorithm, there are limits to the amount of ordering that can be done in this manner.” The stepwise nature of these scope mechanisms makes it hard to state the factors which influence the preference for one quantifier to take scope over another.

Those NLP systems which have managed to provide some sort of account of quantifier scope preferences have done so by using a separate system of heuristics (or *scope critics*) which apply post-syntactically to determine the most likely scoping. LUNAR (Woods, 1986), TEAM (Martin, Appelt, and Pereira, 1986), and the SRI Core Language Engine as described by Moran (1988; 1992) all employ scope rules of this sort. By and large, these rules are of an ad hoc nature, implementing a linguist’s intuitive idea of what factors determine scope possibilities, and no results have been published regarding the accuracy of these methods. For example, Moran (1988) incorporates rules from other NLP systems and from VanLehn (1978), such as a preference for a logically weaker interpretation, the tendency for *each* to take wide scope, and a ban on raising a quantifier across multiple major clause boundaries. The testing of his system is “limited to checking conformance to the stated rules” (pp. 40-41). In addition, these systems are generally incapable of handling unrestricted text such as that found in the Wall Street Journal corpus in a

robust way, because they need to do a full semantic analysis of a sentence in order to make scope predictions. The statistical basis of our model offers increased robustness and the possibility of more serious evaluation on the basis of corpus data.

#### 4 Modeling quantifier scope

In this section, we argue for an empirically-driven, machine learning approach to the identification of factors relevant to quantifier scope, and the modeling of scope preferences. Following much recent work which applies the tools of machine learning to linguistic problems (Brill, 1995; Pedersen, 2000; van Halteren, Zavrel, and Daelemans, 2001; Soon, Ng, and Lim, 2001), we will treat the prediction of quantifier scope as an example of a *classification* task. Our aim is to provide a robust model of scope prediction based on KT&W's theoretical foundation, and to address the serious lack of empirical results regarding quantifier scope in computational work. We describe here the modeling tools borrowed from the field of Artificial Intelligence for the scope prediction task, and the data from which the generalizations are to be learned. Finally, we present the results of training different incarnations of our scope module on the data, and assess the implications of this exercise for theoretical and computational linguistics.

##### 4.1 Classification in machine learning

Determining which quantifier takes wide scope, given a number of different sources of evidence is an example of what is known in Machine Learning as a classification task (Mitchell, 1996). There are many types of classifiers which may be applied to this task, which both are more sophisticated than the approach suggested by KT&W, and have a more solid probabilistic foundation. These include the Naïve Bayes classifier (Manning and Schütze, 1999; Jurafsky and Martin, 2000), maximum-entropy models (Berger, Della Pietra, and Della Pietra, 1996; Ratnaparkhi, 1997), and the single-layer perceptron

(Bishop, 1995) which we employ here primarily because of their straightforward probabilistic interpretation and their similarity to the scope model of KT&W (since they each could be said to implement a kind of weighted voting of factors). In Section 4.3 below, we describe how classifiers of these types can be constructed to serve as a grammatical module responsible for quantifier scope determination.

All of these classifiers can be trained in a supervised manner. That is, given a sample of training data which provides all of the information which is deemed to be relevant to quantifier scope, and the actual scope reading assigned to a sentence, these classifiers will attempt to extract generalizations which can be fruitfully applied in classifying as yet unseen examples.

## 4.2 Data

The data on which the quantifier scope classifiers are trained and tested is an extract from the Penn Treebank (Marcus, Santorini, and Marcinkiewicz, 1993) which we have tagged to indicate the most salient scope interpretation of each sentence in context. Figure 3 shows an example of a training sentence with the scope reading indicated. The lower quantifier bears the tag ‘Q1’, and the higher quantifier bears the tag ‘Q2’, so this sentence is interpreted such that the lower quantified expression has wide scope. Reversing the tags would have meant that the higher quantifier takes wide scope, while if both quantifiers had been marked with ‘Q1’, this would have indicated that there is no scope interaction between them (as when they are logically independent, or take scope in different conjuncts of a conjoined phrase).<sup>2</sup>

The sentences tagged were chosen from the Wall Street Journal section of the Tree-

---

<sup>2</sup> This “no interaction” class is a sort of “elsewhere” category which results from phrasing the classification question as “Which quantifier takes wider scope in the preferred reading?”. Where there is no scope interaction, the answer is “neither”. This includes cases in which the relative scope of operators does not correspond to a difference in meaning, as in *One woman bought one horse*, or when they take scope in different propositional domains, such as in *Mary bought two horses and sold three sheep*.

**Figure 3**

Tagged Wall Street Journal text from the Penn Treebank. The lower quantified expression takes wide scope, indicated by its tag 'Q1'.

```
( (S
  (NP-SBJ
    (NP (DT Those) )
    (SBAR
      (WHNP-1 (WP who) )
      (S
        (NP-SBJ-2 (-NONE- *T*-1) )
        (ADVP (RB still) )
        (VP (VBP want)
          (S
            (NP-SBJ (-NONE- *-2) )
            (VP (TO to)
              (VP (VB do)
                (NP (PRP it) ))))))))
        ('' ''
          (VP (MD will)
            (ADVP (RB just) )
            (VP (VB find)
              (NP
                (NP (DT-Q2 some) (NN way) )
                (SBAR
                  (WHADVP-3 (-NONE- 0) )
                  (S
                    (NP-SBJ (-NONE- *) )
                    (VP (TO to)
                      (VP (VB get)
                        (PP (IN around) ('' ''
                          (NP (DT-Q1 any) (NN attempt)
                            (S
                              (NP-SBJ (-NONE- *) )
                              (VP (TO to)
                                (VP (VB curb)
                                  (NP (PRP it) ))))))
                              (ADVP-MNR (-NONE- *T*-3) ))))))))
                    (NP (PRP it) ))))))))
          (. .) ))
```

bank to have a certain set of attributes which simplify the task of designing the quantifier scope module of the grammar. First, in order to simplify the coding process, each sentence has exactly two scope-taking elements of the sort considered for this project.<sup>3</sup> These include most noun phrases which begin with a determiner, predeterminer, or QP<sup>4</sup>, but exclude NPs in which the determiner is *a*, *an* or *the*. Excluding these determiners from

<sup>3</sup> This restriction that each sentence contain only two quantified elements does not actually exclude many sentences from consideration. We identified only 61 sentences with three quantifiers of the sort we consider, and 12 sentences with four. In addition, our review of these sentences revealed that many of them simply involve lists, in which the quantifiers do not interact scopally (as in, for example, "We ask that you turn off all cell phones, extinguish all cigarettes, and open any candy before the performance begins"). Thus, the class of sentences with more than two quantifiers is small, and seems to involve even simpler quantifier interactions than those found in our corpus.

<sup>4</sup> These categories are intended to be understood as they are used in the tagging and parsing of the Penn Treebank. See Santorini (1990) and Bies et al. (1995) for details. The category QP is particularly unintuitive in that it does not correspond to a quantified noun phrase, but to a measure expression, such as *more than half*.

consideration largely avoids the problem of generics and the complexities of assigning scope readings to definite descriptions. In addition, only sentences which had the root node S were considered. This serves to exclude sentence fragments and interrogative sentence types. Our data set therefore differs systematically from the full WSJ corpus, but we believe it is sufficient to allow many generalizations about English quantification to be induced. Given these restrictions on the input data, the task of the scope classifier is a choice between three alternatives<sup>5</sup>:

there is no scopal interaction	(class 0)
the first quantifier takes wide scope	(class 1)
the second quantifier takes wide scope	(class 2)

The result is a set of 893 sentences<sup>6</sup>, annotated with Penn Treebank II parse trees and hand-tagged for the primary scope reading. To assess the reliability of the hand-tagged data used in this project, the data were coded a second time, in addition to the reference coding. The independent codings agreed on 76.3% of sentences. The kappa statistic (Cohen, 1960) for agreement was .52, with a 95% confidence interval between [.40, .64]. Krippendorff (1980) has been widely cited as advocating the view that  $\kappa > .8$  should be taken as good reliability, and  $.67 < \kappa < .8$  tentative reliability, but we are satisfied with the level of inter-coder agreement on this task. As Carletta (1996) notes, many tasks in computational linguistics are simply more difficult than the content analysis classifications addressed by Krippendorff, and according to Fleiss (1981),  $.4 < \kappa < .75$  indicates fair to good agreement anyhow. Because of our relatively small amount of data, we use the technique of ten-fold cross-validation in evaluating our classifiers, in each case choosing 89 sentences from the data as a test set, and training on the rest.

We pre-processed the data in order to extract the information from each sentence

---

<sup>5</sup> Some linguists may find it strange that we have chosen to treat the choice of preferred scoping for two quantified elements as a tripartite decision, since the possibility of independence is seldom treated in the linguistic literature. As we are dealing with corpus data in this experiment, we cannot afford to ignore this possibility.

<sup>6</sup> These data have been made publicly available to all licensees of the Penn Treebank by means of a patchfile which may be retrieved from <http://home.uchicago.edu/~dchiggin/qscope-data.tgz>.

which we would be treating as relevant to the prediction of quantifier scoping in this project. (Although the initial coding of the preferred scope reading for a sentence was done manually, this reorganization of the data was done automatically.) At the end of this procedure, each sentence was represented as a record containing the following information:

- the syntactic category, according to Penn Treebank conventions, of the first quantifier (e.g., DT for *each*, NN for *everyone*, or QP for *more than half*)
- the first quantifier as a lexical item (e.g., *each* or *everyone*). For a QP consisting of multiple words, this field contains the head word, or ‘CD’ in case the head is a cardinal number.
- the syntactic category of the second quantifier
- the second quantifier as a lexical item
- the syntactic category of the lowest node dominating both quantified NPs (the “join” node)
- whether the first quantified NP c-commands the second
- whether the second quantified NP c-commands the first
- the number of nodes intervening<sup>7</sup> between the two quantified NPs
- a list of the different categories of nodes which intervene between the quantified NPs (actually, for each non-terminal category, there is a distinct binary feature indicating whether a node of that category intervenes.)

---

<sup>7</sup> We take a node  $\alpha$  to intervene between two other nodes  $\beta$  and  $\gamma$  in a tree if and only if

- i  $\delta$  is the lowest node dominating both  $\beta$  and  $\gamma$ ,
- ii  $\delta$  dominates  $\alpha$  or  $\delta = \alpha$ , and
- iii  $\alpha$  dominates either  $\beta$  or  $\gamma$ .

**Figure 4**  
Example record corresponding to the sentence shown in Figure 3

class:	2
first cat:	DT
first head:	some
second cat:	DT
second head:	any
join cat:	NP
first c-commands:	YES
second c-commands:	NO
nodes intervening:	6
VP intervenes:	YES
ADVP intervenes:	NO
⋮	
S intervenes:	YES
conj intervenes:	NO
, intervenes:	NO
: intervenes:	NO
⋮	
” intervenes:	YES

- whether a conjoined node intervenes between the quantified NPs
- a list of the punctuation types which are immediately dominated by nodes intervening between the two NPs (again, for each punctuation tag in the treebank there is a distinct binary feature indicating whether such punctuation intervenes.)

Figure 4 illustrates how these features would be used to encode the example in Figure 3.

These are not the exact factors which KT&W (1999) suggest be taken into consideration in making scope predictions, and they are certainly not sufficient to completely determine the proper scope reading for all sentences. Surely pragmatic factors and real-world knowledge influence our interpretations as well, although these are not represented here. However, this list does provide information which could potentially be useful in predicting the best scope reading for a sentence. For example, information about whether one quantified NP c-commands the other corresponds to KT&W’s observation that subject quantifiers tend to take wide scope over object quantifiers, and topicalized quantifiers tend to outscope everything. The identity of each lexical quantifier clearly should allow

**Table 2**

Baseline performance, summed over all ten test sets

Condition	Correct	Incorrect	Percentage Correct
First has wide scope	0	64	$0/64 = 0.0\%$
Second has wide scope	0	281	$0/281 = 0.0\%$
No scope interaction	545	0	$545/545 = 100.0\%$
Total	545	345	$545/890 = 61.2\%$

our classifiers to make the generalization that *each* tends to take wide scope, if this is found in the data, and perhaps even KT&W’s observation that universal quantifiers tend to outscope existentials.

### 4.3 Classifier design

In this section, we present the three models which we have trained to predict the preferred quantifier scoping on Penn Treebank sentences: a Naïve Bayes classifier, a maximum-entropy classifier, and a single-layer perceptron.<sup>8</sup> In evaluating how well these models do in assigning the proper scope reading to each test sentence, it is important to have a baseline for comparison. The baseline model for this task is one which simply guesses the most frequent category of the data (“no scope interaction”) every time. This simplistic strategy already classifies 61.2% of the test examples correctly, as shown in Table 2.

It may surprise some linguists that this third class of sentences in which there is no scopal interaction between the two quantifiers is the largest. In part, this may be due to special features of the Wall Street Journal text of which the corpus consists. For example, newspaper articles may contain more direct quotations than other genres. However, in the process of tagging the data it was also apparent that in a large proportion of cases, the two quantifiers were taking scope in different conjuncts of a conjoined phrase. This further tendency supports the idea that people may intentionally avoid constructions in

<sup>8</sup> The implementations of these classifiers are publicly available as Perl modules at <http://home.uchicago.edu/~dchiggin/classifiers.tgz>.

which there is even the possibility for quantifier scope interactions—perhaps due to some hearer-oriented pragmatic principle. Linguists may also be concerned that this additional category in which there is no scope interaction between quantifiers makes it difficult to compare the results of the present work with theoretical accounts of quantifier scope which ignore this case and concentrate on instances in which one quantifier does take scope over another. Firstly, however, we provide a model of scope prediction rather than scope generation, and so it is in any case not directly comparable with work in theoretical linguistics which has largely ignored scope preferences. Second, the empirical nature of this study requires that we take note of cases in which the quantifiers simply do not interact.

**4.3.1 Naïve Bayes** Our data  $D$  will consist of a vector of features  $(d_0 \dots d_n)$ , which represent aspects of the sentence such as whether one quantified expression c-commands the other, as described in Section 4.2 above. The fundamental simplifying assumption which we make in designing a Naïve Bayes classifier is that these features are independent of one another, and therefore can be aggregated as independent sources of evidence about which class  $c^*$  a given sentence belongs to. This independence assumption is formalized in Equations 1 and 2.

$$c^* = \arg \max_c P(c)P(d_0 \dots d_n|c) \quad (1)$$

$$\approx \arg \max_c P(c) \prod_{k=0}^n P(d_k|c) \quad (2)$$

We constructed an empirical estimate of the prior probability  $P(c)$  by simply counting the frequency with which each class occurs in the training data. We constructed each  $P(d_k|c)$  by counting how often each feature  $d_k$  co-occurs with the class  $c$ , to construct the empirical estimate  $\hat{P}(d_k|c)$ , and interpolated this with the empirical frequency  $\hat{P}(d_k)$

**Table 3**

Performance of the Naïve Bayes classifier, summed over all ten test runs

Condition	Correct	Incorrect	Percentage Correct
First has wide scope	177	104	$177/281 = 63.0\%$
Second has wide scope	41	23	$41/64 = 64.1\%$
No scope interaction	428	117	$428/545 = 78.5\%$
Total	646	244	$646/890 = 72.6\%$

of the feature  $d_k$ , not conditioned on the class  $c$ . This interpolated probability model was used in order to smooth the probability distribution, avoiding the problems which can arise if certain feature-value pairs are assigned a probability of zero.

The performance of the Naïve Bayes classifier is summarized in Table 3. For each of the ten test sets of 89 items taken from the corpus, the remaining 804 sentences were used to train the model. The Naïve Bayes classifier outperformed the baseline, and its errors were clearly more evenly distributed across conditions than those of the baseline model.

In addition to the raw counts of test examples correctly classified, though, we would like to know something of the internal structure of the model, i.e., what sort of features it has induced from the data. For this classifier, we can assume that a feature  $f$  is a good predictor of a class  $c^*$  when the value of  $P(f|c^*)$  is significantly larger than the (geometric) mean value of  $P(f|c)$  for all other values of  $c$ . Those features with the greatest ratio  $P(f) \times \frac{P(f|c^*)}{\text{geom.mean}(\forall c \neq c^* [P(f|c)])}$  are listed in Table 4.<sup>9</sup>

The first feature in Table 4 shows that there is a tendency for quantified elements not to interact when they are found in conjoined constituents, while the second feature indicates a preference for quantifiers not to interact when there is an intervening comma (presumably an indicator of greater syntactic “distance”.) Feature 3 indicates a preference

<sup>9</sup> We include the term  $P(f)$  in the product in order to prevent sparsely instantiated features from showing up as highly-ranked.

**Table 4**  
Most active features from Naïve Bayes classifier

Rank	Feature	Predicted class	Ratio
1	There is an intervening conjunct node	0	1.63
2	There is an intervening comma	0	1.51
3	There is an intervening S node	1	1.33
4	The first quantified NP does not c-command the second	0	1.25
5	Second quantifier is tagged QP	1	1.16
6	There is an intervening S node	0	1.12
15	The second quantified NP c-commands the first	2	1.00

for class **1** when there is an intervening S node, while feature 6 indicates a preference for class **0** under the same conditions. Presumably, this reflects a dispreference for the second quantifier to take wide scope when there is a clause boundary intervening between it and the first quantifier. The fourth feature in Table 4 indicates that, if the first quantified NP does not c-command the second, it is less likely to take wide scope. This is not surprising, given the importance which c-command relations have had in theoretical discussions of quantifier scope. The fifth feature expresses a preference for quantified expressions of category QP to take narrow scope, if they are the second of the two quantifiers under consideration. This may simply be reflective of the fact that class **1** is more common than class **2**, and the measure expressions found in QP phrases in the Penn Treebank (such as *more than three*, or *about half*) tend not to be logically independent of other quantifiers. Finally, the last feature in Table 4 indicates a high correlation between the second quantified expression c-commanding the first, and the second quantifier taking wide scope. We can easily see this as a translation into our feature set of KT&W’s claim that subjects tend to out-scope objects and obliques, and topicalized elements tend to take wide scope. Some of these top-ranked features have to do with information only found in the written medium, but on the whole, the features induced by the Naïve Bayes classifier seem consistent with those suggested by KT&W, although they are distinct by necessity.

**4.3.2 Maximum entropy** The maximum-entropy classifier is a sort of loglinear model, defining the joint probability of a class and a data vector  $(d_0 \dots d_n)$  as the product of the prior probability of the class  $c$  with a set of features related to the data:<sup>10</sup>

$$P(d_0 \dots d_n, c) = \frac{P(c)}{Z} \prod_{k=0}^n \alpha_k \quad (3)$$

---

<sup>10</sup>  $Z$  in Equation 3 is simply a normalizing constant which ensures that we end up with a probability distribution.

**Table 5**

Performance of the maximum-entropy classifier, summed over all ten test runs

Condition	Correct	Incorrect	Percentage Correct
First has wide scope	148	133	148/281 = 52.7%
Second has wide scope	31	33	31/64 = 48.4%
No scope interaction	475	70	475/545 = 87.2%
Total	654	236	654/890 = 73.5%

This superficially resembles the form of the Naïve Bayes classifier in Equation 2, but it differs in that the way in which values for each  $\alpha$  are chosen does not assume that the features in the data are independent. For each of the ten training sets, we used the Generalized Iterative Scaling algorithm to train this classifier on 654 training examples, using 150 examples for validation in order to choose the best set of values for the  $\alpha$ 's.<sup>11</sup> Test data could then be classified by choosing the class for the data which maximizes the joint probability in Equation 3.

The results are shown in Table 5. The maximum-entropy classifier showed slightly higher performance than the Naïve Bayes classifier, with the lowest error rate on the class of sentences having no scope interaction.

To determine exactly which features of the data the maximum entropy classifier sees as relevant to the classification problem, we can simply look at the  $\alpha$  values (from Equation 3) for each feature. Those features with higher values for  $\alpha$  are weighted more heavily in determining the proper scoping. Some of the features with the highest values for  $\alpha$  are listed in Table 6. Because of the way the classifier is built, predictor features for class 2 need to have higher loadings in order to overcome the lower prior probability of the class. Therefore, we actually rank the features in Table 6 according to  $\alpha \hat{P}(c)^k$  (which we denote as  $\alpha_{c,k}$ ).  $\hat{P}(c)$  represents the empirical prior probability of a class  $c$ , and  $k$  is

<sup>11</sup> Overtraining is not a problem with the pure version of the Generalized Iterative Scaling algorithm. However, for efficiency reasons we chose to take the training corpus as representative of the event space, rather than enumerating the space exhaustively (see Jelinek (1998) for details). For this reason, it was necessary to employ validation in training.

**Table 6**  
Top feature loadings from maximum entropy classifier

Rank	Feature	Predicted class	$\alpha_{c,.25}$
1	Second quantifier is <i>each</i>	2	1.13
2	There is an intervening comma	0	1.01
3	There is an intervening conjunct node	0	1.00
4	First quantified NP does not c-command the second	0	0.99
5	Second quantifier is <i>every</i>	2	0.98
6	There is an intervening quotation mark (")	0	0.95
7	There is an intervening colon	0	0.95
12	First quantified NP c-commands the second	1	0.92
25	There is no intervening comma	1	0.90

simply a constant (.25 in this case) chosen in order to try to get a mix of features for different classes at the top of the list.

The features ranked first and fifth in Table 6 express lexical preferences for certain quantifiers to take wide scope, even when they are the second of the two quantifiers according to linear order in the string of words. The tendency for *each* to take wide scope is stronger than for the other quantifier, which is in line with KT&W’s decision to list it as the only quantifier with a lexical preference for scoping. Feature 2 makes the “no scope interaction” class more likely if a comma intervenes, while Feature 25 makes a wide-scope reading for the first quantifier more likely if there is no intervening comma. The third feature expresses the tendency mentioned above for quantifiers in conjoined clauses not to interact. Features 4 and 12 indicate that if the first quantified expression c-commands the second, it is likely to take wide scope, and if this is not the case there is likely to be no scope interaction. Finally, the sixth and seventh features in the table show that an intervening quotation mark or colon will make the classifier tend toward class 0, “no scope interaction”, which is easy to understand. Quotations are often opaque to quantifier scope interactions. The top features found by the maximum-entropy classifier largely coincide with those found by the Naïve Bayes model, which indicates that these generalizations are robust and objectively present in the data.

**4.3.3 Single-layer perceptron** For our neural network classifier, we used a feed-forward single-layer perceptron, with the **softmax** function used to determine the activation of nodes at the output layer, because this is a 1-of-n classification task (Bridle, 1990). The data to be classified is presented as a vector of features at the input layer, and the output layer has three nodes, representing the three possible classes for the data: “first has wide scope”, “second has wide scope”, and “no scope interaction”. The output node with the highest activation is interpreted as the class of the datum presented at the input layer.

For each of the ten test sets of 89 examples, we trained the connection weights of the network using error back-propagation on 654 training sentences, reserving 150 sentences for validation in order to choose the weights from the training epoch with the highest classification performance. In Table 7 we present the results of the single-layer neural network in classifying our test sentences. The single-layer perceptron has much better classification performance than the Naïve Bayes classifier and maximum-entropy model. This may be because the training of the network aims to minimize error in the activation of the classification output nodes, which is directly related to the classification task at hand, whereas the other models do not directly make use of the notion of “classification error”. The perceptron also uses a sort of weighted voting, and could be interpreted as an implementation of KT&W’s proposal for scope determination. This clearly illustrates that the tenability of their proposal hinges on the exact details of its implementation, since all of our classifier models are reasonable interpretations of their approach, but have very different performance results on our scope determination task.

To determine exactly which features of the data the network sees as relevant to the classification problem, we can simply look at the connection weights for each feature/class pair. Higher connection weights indicate a greater correlation between input features and output classes. For one of the ten networks we trained, some of the features with the

---

**Table 7**  
Performance of the single-layer perceptron, summed over all ten test runs

---

Condition	Correct	Incorrect	Percentage Correct
First has wide scope	182	111	$182/281 = 64.8\%$
Second has wide scope	35	29	$35/64 = 54.7\%$
No scope interaction	468	77	$468/545 = 85.9\%$
Total	685	205	$685/890 = 77.0\%$

**Table 8**  
Most active features from single-layer perceptron

Rank	Feature	Predicted class	Weight
1	There is an intervening comma	0	4.31
2	Second quantifier is <i>all</i>	0	3.77
3	There is an intervening colon	0	2.98
4	There is an intervening conjunct node	0	2.72
17	The first quantified NP c-commands the second	1	1.69
18	Second quantifier is tagged RBS	2	1.69
19	There is an intervening S node	1	1.61
20	Second quantifier is <i>each</i>	2	1.50

highest connection weights are listed in Table 8. Since class 0 is simply more frequent in the training data than the other two classes, the weights for this class tend to be higher. Therefore, we also list some of the best predictor features for classes 1 and 2 in the table.

The first and third features in Table 8 show that an intervening comma or colon will make the classifier tend toward class 0, “no scope interaction”. This is similar to the relevance of an intervening quotation mark found by the maximum-entropy classifier, and can be taken as an indication that quantifiers in distant syntactic subdomains are unlikely to interact. Similarly, the fourth feature indicates that quantifiers in separate conjuncts are unlikely to interact. The second feature in the table expresses a tendency for there to be no scope interaction between two quantifiers if the second of them is headed by *all*. This may be related to the independence of universal quantifiers ( $\forall x \forall y [\phi] \iff \forall y \forall x [\phi]$ ). The 17th-ranked feature in Table 8 indicates a high correlation between the first quantified expression c-commanding the second, and the first quantifier taking wide scope, which again supports KT&W’s claim that scope preferences are related to syntactic superiority relations. The 18th-ranked feature expresses a preference for a quantified expression headed by *most* to take wide scope, even if it is the second of the two quantifiers (since *most* is the only quantifier in the corpus which bears the tag RBS). Feature 19 indicates that the first quantifier is more likely to take wide scope if there is a clause

**Table 9**  
Summary of classifier results

	Training Data	Validation Data	Test Data
Baseline	–	–	61.2%
Naïve Bayes	76.7%	–	72.6%
Maximum entropy	78.3%	75.5%	73.5%
Single-layer perceptron	84.7%	76.8%	77.0%

boundary intervening between the two quantifiers, which supports the notion that the syntactic distance between the quantifiers is relevant to scope preferences. Finally, feature 20 expresses the well-known tendency for quantified expressions headed by *each* to take wide scope.

#### 4.4 Summary of results

Table 9 summarizes the performance of the quantifier scope models we have presented here. All of the classifiers have test set accuracy above the baseline, which a paired t-test reveals to be significant at the .001 level, while the differences between the Naïve Bayes, maximum-entropy, and single-layer perceptron classifiers are not statistically significant.

These three classifiers directly implement a sort of weighted voting, the method of aggregating evidence proposed by KT&W (although they are slightly more sophisticated than the unweighted voting which is actually used in their paper). Of course, since we do not use exactly the set of features suggested by KT&W, our model should not be seen as a straightforward implementation of the theory outlined in their 1999 paper. Nevertheless, the results in Table 9 suggest that KT&W’s suggested design can be used with some success in modeling scope preferences. Moreover, this project provides an answer to some of the objections which Aoun and Li (2000) raise to KT&W. Aoun & Li claim that KT&W’s choice of experts is seemingly arbitrary, and that it is unclear how the voting weights of each expert are to be set, but the machine learning approach

we employ in this paper is capable of addressing both potential problems. Supervised training of our classifiers is a straightforward approach to setting the weights, and also constitutes our approach to selecting features, or “experts” in KT&W’s terminology. In the training process, any feature which is irrelevant to scoping preferences should receive weights which make its effect negligible.

## 5 Syntax and Scope

In this section, we show how the classifier models of quantifier scope determination introduced in Section 4 may be integrated with a PCFG model of syntax. We compare two different ways in which the two components may be combined, which may loosely be termed *serial* and *parallel*, and argue for the latter on the basis of empirical results.

### 5.1 Modular design

Our use of a phrase-structure syntactic component and a quantifier-scope component to define a combined language model is simplified by the fact that our classifiers are probabilistic, and define a conditional probability distribution over quantifier scopings. The probability distributions which our classifiers define for quantifier scope structures are conditional on syntactic phrase structure, because they are computed on the basis of syntactically provided features, such as the number of nodes of a certain type which intervene between two quantifiers in a phrase structure tree.

Thus, the combined language model which we define in this chapter assigns probabilities according to the pairs of structures a sentence may be assigned by the Q-Structure and phrase-structure syntax modules. The probability of a word string  $w_{1-n}$  is therefore defined as in Equation 4, where  $Q$  ranges over all possible Q-Structures in the set  $\mathcal{Q}$ , and  $S$  ranges over all possible syntactic structures in the set  $\mathcal{S}$ .

$$P(w_{1-n}) = \sum_{S \in \mathcal{S}, Q \in \mathcal{Q}} P(S, Q | w_{1-n}) \quad (4)$$

$$= \sum_{S \in \mathcal{S}, Q \in \mathcal{Q}} P(S | w_{1-n}) P(Q | S, w_{1-n}) \quad (5)$$

Equation 5 shows how we can use the definition of conditional probability to break our calculation of the language model probability into two parts. The first part,  $P(S | w_{1-n})$ , which we may abbreviate as simply  $P(S)$ , is the probability of a syntactic tree structure being assigned to a word string. We model this using a probabilistic phrase-structure grammar (cf. Charniak (1993), Charniak (1997), Collins (1997)). The second distribution on the right side of Equation 5 is the conditional probability of a particular quantifier-scope structure being assigned to a word string, given the syntactic structure of that string. This probability is written as  $P(Q | S, w_{1-n})$ , or simply  $P(Q | S)$ , and represents the quantity we estimated above in constructing classifiers to predict the scopal representation of a sentence based on aspects of its syntactic structure.

Thus, given a PCFG model of syntactic structure, and a probabilistically-defined classifier of the sort introduced in Section 4, it is simple to determine the probability of any pairing of two structures from each domain for a given sentence. We simply multiply the values of  $P(S)$  and  $P(Q | S)$  to obtain the joint probability  $P(Q, S)$ . In the current section, we examine two different models of combination for these components: one in which scope determination is applied to the optimal syntactic structure (the Viterbi parse), and one in which optimization is done in the space of both modules to find the optimal pairing of syntactic and quantifier-scope structures.

## 5.2 The syntactic module

Before turning to the application of our multi-modular approach to the problem of scope determination in Section 5.3, we present here a short overview of the phrase-structure

syntactic component used in these projects. As noted above, we model syntax as a probabilistic phrase-structure grammar (PCFG), and in particular, we use a *treebank grammar* (Charniak, 1996) trained on the Penn Treebank.

A PCFG defines the probability of a string of words as the sum of the probabilities of all admissible phrase-structure parses (trees) for that string. The probability of a tree is the product of the probability of all of the rule instances used in the construction of that tree, where rules take the form  $N \rightarrow \phi$ ,  $N$  a nonterminal symbol and  $\phi$  a finite sequence of one or more terminals or nonterminals.

To take an example, Figure 6 illustrates a phrase-structure tree for the sentence *Susan might not believe you*, which is admissible according to the grammar in Figure 5. (All of the minimal subtrees in Figure 6 are instances of one of our rules.) The probability of this tree, which we can indicate as  $\tau$ , can be calculated as in (6).

$$P(\tau) = \prod_{\rho \in Rules(\tau)} P(\rho) \quad (6)$$

$$\begin{aligned} &= P(S \rightarrow NP VP) \times P(VP \rightarrow V VP) \times P(VP \rightarrow ADV VP) \\ &\quad \times P(VP \rightarrow V NP) \times P(NP \rightarrow Susan) \times P(V \rightarrow might) \\ &\quad \times P(ADV \rightarrow not) \times P(V \rightarrow believe) \times P(NP \rightarrow you) \end{aligned} \quad (7)$$

$$= .7 \times .3 \times .1 \times .3 \times .3 \times .2 \times .5 \times .3 \times .4 = 2.268 \times 10^{-5} \quad (8)$$

The actual grammar rules and associated probabilities which we use in defining our syntactic module are derived from the WSJ corpus of the Penn Treebank by maximum-likelihood estimation. That is, for each rule  $N \rightarrow \phi$  used in the treebank, we add the rule to the grammar, and set its probability to  $\frac{C(N \rightarrow \phi)}{\sum_{\psi} C(N \rightarrow \psi)}$ , where  $C(\cdot)$  denotes the “count” or a rule, i.e., the number of times it is used in the corpus. A grammar composed in this manner is referred to as a treebank grammar, because its rules are directly derived from those in a treebank corpus.

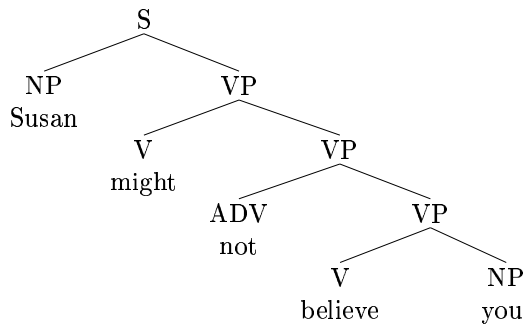
**Figure 5**

A simple probabilistic phrase-structure grammar.

Rule	Probability
S → NP VP	.7
S → VP	.2
S → V NP VP	.1
VP → V VP	.3
VP → ADV VP	.1
VP → V	.1
VP → V NP	.3
VP → V NP NP	.2
NP → Susan	.3
NP → you	.4
NP → Yves	.3
V → might	.2
V → believe	.3
V → show	.3
V → stay	.2
ADV → not	.5
ADV → always	.5

**Figure 6**

A simple phrase-structure tree.



We used sections 00–20 of the WSJ corpus of the Penn Treebank for collecting the rules and associated probabilities of our PCFG, which is implemented as a bottom-up chart-parser. Before constructing the grammar, the treebank was pre-processed using known procedures (cf. Krotov et al. (1998), Belz (2001)) to facilitate the construction of a rule list. Functional and anaphoric annotations (basically anything following a “-” in a node label; cf. Santorini (1990), Bies et al. (1995)) were removed from nonterminal labels. Nodes which dominate only “empty categories” such as traces were removed. In addition, unary-branching constructions were removed by replacing the mother category in such a structure with the daughter node. (For example, given an instance of the rule  $X \rightarrow YZ$ , if the daughter category  $Y$  were expanded by the unary rule  $Y \rightarrow W$ , our algorithm would induce the single rule  $X \rightarrow WZ$ .) Finally, we discarded all rules which had more than ten symbols on the right-hand side (an arbitrary limit of our parser implementation). This resulted in a total of 18,820 rules, of which 11,156 were discarded as hapax legomena, leaving 7,664 rules in our treebank grammar. Figure 7 shows some of the rules in our grammar with the highest and lowest corpus counts.

### 5.3 Unlabeled scope determination

In this section, we describe an experiment designed to assess the performance parallel and serial approaches to combining grammatical modules, focusing on the task of *unlabeled scope determination*. This task involves predicting the most likely Q-Structure representation for a sentence, basically the same task we addressed in Section 4, in comparing the performance levels of each type of classifier. However, the experiment of this section differs in that instead of providing a syntactic tree from the Penn Treebank as input to the classifier, we provide the model only with a string of words (a sentence). Our dual-component model will search for the optimal syntactic and scopal structures for the sentence (the pairing  $(\tau^*, \chi^*)$ ), and will be evaluated based on its success in identifying

**Figure 7**

Rules derived from sections 00–20 of the Penn Treebank WSJ corpus. “TOP” is a special “start” symbol which may expand to any of the symbols found at the root of a tree in the corpus.

Rule	Corpus Count
PP → IN NP	59053
TOP → S	34614
NP → DT NN	28074
NP → NP PP	25192
S → NP VP	14032
S → NP VP .	12901
VP → TO VP	11598
⋮	
S → CC PP NNP NNP VP .	2
NP → DT “ NN NN NN ”	2
NP → NP PP PP PP PP PP	2
INTJ → UH UH	2
NP → DT “ NN NNS	2
SBARQ → “ WP VP . ”	2
S → PP NP VP . ”	2

the correct scope reading  $\chi^*$ .

Our concern in this section will be to determine whether it is necessary to search the space of possible pairings  $(\tau, \chi)$  of syntactic and scopal structures, or whether it is sufficient to use our PCFG to first fix the syntactic tree  $\tau$ , and then to choose a scope reading to maximize the probability of the pairing. That is, are syntax and quantifier scope mutually dependent components of grammar, or can scope relations be “read off” the syntax? The serial model suggests that the optimal syntactic structure  $\tau^*$  should be chosen on the basis of the syntactic module only, as in Equation 9, and the optimal quantifier-scope structure  $\chi^*$  then chosen on the basis of  $\tau^*$ , as in Equation 10. The parallel model, on the other hand, suggests that the most likely pairing of structures must be chosen in the joint probability space of both components, as in Equation 11.

$$\tau^* = \arg \max_{\tau \in \mathcal{S}} P_S(\tau | w_{1-n}) \quad (9)$$

$$\chi^* = \arg \max_{\chi \in \mathcal{Q}} P_Q(\chi | \tau^*, w_{1-n}) \quad (10)$$

$$\tau^* = \{ \tau \mid (\tau, \chi) = \arg \max_{\tau \in \mathcal{S}, \chi \in \mathcal{Q}} P_S(\tau | w_{1-n}) P_Q(\chi | \tau, w_{1-n}) \} \quad (11)$$

**5.3.1 Experimental design** For this experiment, we implement the scoping component as a single-layer feed-forward network, because that classifier had the best prediction rate among all classifiers. The **softmax** activation function we use for the output nodes of the classifier guarantees that the activations of all of the output nodes sum to one, and can be interpreted as class probabilities. The syntactic component, of course, is determined by the treebank PCFG grammar described above.

Given these two models, which respectively define  $P_Q(\chi|\tau, w_{1-n})$  and  $P_S(\tau|w_{1-n})$  from Equation 11 above, it only remains to specify how to search the space of pairings  $(\tau, \chi)$  in doing this optimization to find  $\chi^*$ . Unfortunately, it is not feasible to examine all values  $\tau \in \mathcal{S}$ , since our PCFG will generally admit a huge number of trees for a sentence (especially given a mean sentence length of over 20 words in the WSJ corpus).<sup>12</sup> Our solution to this search problem is to make the simplifying assumption that the syntactic tree which is used in the optimal set of structures  $(\tau^*, \chi^*)$  will always be among the top few trees  $\tau$  for which  $P_S(\tau|w_{1-n})$  is the greatest. That is, while we suppose that quantifier scope information is relevant to parsing, we do not suppose that it is so strong a determinant as to completely override syntactic factors. In practice, this means that our parser will return the top 10 parses for each sentence, along with the probabilities they are assigned, and these are the only parses which are considered in looking for the optimal set of linguistic structures.

We again used ten-fold cross-validation in evaluating the models, dividing the scope-tagged corpus into ten test sections of 89 sentences each, and we used the same version of the treebank grammar for our PCFG. The first model retrieved the top ten syntactic parses  $(\tau_0 \cdots \tau_9)$  for each sentence, and computed the probability  $P(\tau, \chi)$  for each  $\tau \in \tau_0 \cdots \tau_9, \chi \in \mathbf{0}, \mathbf{1}, \mathbf{2}$ , choosing that scopal representation  $\chi$  which was found in the maximum-probability pairing. We call this the parallel model, because the properties of each probabilistic model may influence the optimal structure chosen by the other. The second model retrieved only the Viterbi parse  $\tau_0$  from the PCFG, and chose the scopal representation  $\chi$  for which the pairing  $(\tau_0, \chi)$  took on the highest probability. We call this the serial model, because it represents syntactic phrase structure as independent of other components of grammar (in this case, quantifier scope), though other components

---

<sup>12</sup> However, since we are only allowing  $\chi$  to range over the three scope readings  $(\mathbf{0}, \mathbf{1}, \mathbf{2})$ , it is possible to enumerate all values of  $\chi$  to be paired with a given syntactic tree  $\tau$ .

**Table 10**

Performance of models on the unlabeled scope prediction task, summed over all ten test runs

Condition	Correct	Incorrect	Percentage Correct
Parallel Model:			
First has wide scope	168	113	167/281 = 59.4%
Second has wide scope	26	38	26/64 = 40.6%
No scope interaction	467	178	467/545 = 85.7%
Total	661	229	661/890 = 74.3%
Serial Model:			
First has wide scope	163	118	163/281 = 58.0%
Second has wide scope	27	37	27/64 = 42.2%
No scope interaction	461	184	461/545 = 84.6%
Total	651	239	651/890 = 73.1%

are dependent upon it.

**5.3.2 Results** There was an appreciable difference in performance between these two models on the quantifier-scope test sets. As shown in Table 10, the parallel model narrowly outperformed the serial model, by 1.2%. A ten-fold paired t-test on the test sections of the scope-tagged corpus shows that the parallel model is significantly better ( $p < .05$ ).

This result suggests that we must take other aspects of structure into account in determining the syntactic structure of a sentence. Searching both dimensions of the hypothesis space for our dual-component model allows the composite model to handle the interdependencies between different aspects of grammatical structure, while fixing a phrase-structure tree purely on the basis of syntactic considerations led to suboptimal performance in using that structure as a basis for determining quantifier scope.

## 6 Conclusion

In this paper, we have taken a statistical, corpus-based approach to the modeling of quantifier scope preferences, a subject which had previously been addressed only with systems of ad hoc rules derived from linguists' intuitive judgements. Our model takes

its theoretical inspiration from Kuno, Takami, and Wu (1999), who suggest an “expert system” approach to scope preferences, and follows many other projects in the machine learning of natural language which combine information from multiple sources in solving linguistic problems.

Our results are generally supportive of the design which KT&W propose for the quantifier scope component of grammar, and some of the features induced by our models find clear parallels in the factors which KT&W claim to be relevant to scoping. In addition, our final experiment, in which we combine our quantifier scope module with a PCFG model of syntactic phrase-structure, provides evidence for a grammatical architecture in which different aspects of structure mutually constrain one another. This result casts doubt on approaches in which syntactic processing is completed prior to the determination of other grammatical properties of a sentence, such as quantifier scope relations.

## References

- Aoun, Joseph and Yen-hui Audrey Li. 1993. *The Syntax of Scope*. MIT Press, Cambridge, Massachusetts.
- Aoun, Joseph and Yen-hui Audrey Li. 2000. Scope, structure, and expert systems: A reply to Kuno et al. *Language*, 76(1):133–155.
- Belz, Anja. 2001. Optimisation of corpus-derived probabilistic grammars. In *Proceedings of Corpus Linguistics 2001*, pages 46–57.
- Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.
- Bies, Ann, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II style. Technical report, Penn Treebank Project, University of Pennsylvania.
- Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford.
- Bridle, J. S. 1990. Probabilistic interpretation of feedforward classification network outputs with relationships to statistical pattern recognition. In F. Fogelman-Soulie and J. Hérault, editors, *Neurocomputing—Algorithms, Architectures, and Applications*. Springer-Verlag, Berlin, pages 227–236.
- Brill, Eric. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.
- Carden, Guy. 1976. *English Quantifiers: Logical Structure and Linguistic Variation*. Academic Press, New York.
- Carletta, Jean. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.
- Charniak, E. 1993. *Statistical language learning*. The MIT Press, Cambridge, Massachusetts.
- Charniak, Eugene. 1996. Tree-bank grammars. In *AAAI/IAAI, Vol. 2*, pages 1031–1036.
- Charniak, Eugene. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI/IAAI*, pages 598–603.

- Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46.
- Collins, Michael. 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of ACL and EACL '97*, pages 16–23.
- Cooper, Robin. 1983. *Quantification and Syntactic Theory*. Reidel, Dordrecht.
- Fleiss, J. 1981. *Statistical Methods for Rates and Proportions*. John Wiley & Sons.
- Hobbs, Jerry R. and Stuart M. Shieber. 1987. An algorithm for generating quantifier scopings. *Computational Linguistics*, 13:47–63.
- Hornstein, Norbert. 1995. *Logical Form: from GB to Minimalism*. Blackwell, Oxford & Cambridge.
- Jelinek, Frederick. 1998. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts.
- Jurafsky, D. and J. Martin. 2000. *Speech and Language Processing*. Prentice Hall, New Jersey.
- Krippendorff, Klaus. 1980. *Content Analysis: An Introduction to Its Methodology*. Sage Publications.
- Krotov, Alexander, Mark Hepple, Robert J. Gaizauskas, and Yorick Wilks. 1998. Compacting the Penn treebank grammar. In *COLING-ACL*, pages 699–703.
- Kuno, Susumu, Ken-Ichi Takami, and Yuru Wu. 1999. Quantifier scope in English, Chinese, and Japanese. *Language*, 75(1):63–111.
- Kuno, Susumu, Ken-Ichi Takami, and Yuru Wu. 2001. Response to Aoun and Li. *Language*, 77(1):134–143.
- Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.
- Marcus, M., S. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Martin, Paul, Douglas Appelt, and Fernando Pereira. 1986. Transportability and generality in a natural-language interface system. In B. J. Grosz, K. Sparck Jones, and B. L. Webber, editors, *Natural Language Processing*. Kaufmann, Los Altos, CA, pages 585–593.
- May, Robert. 1985. *Logical Form: its Structure and Derivation*. MIT Press, Cambridge, MA.
- McCawley, James D. 1998. *The Syntactic Phenomena of English*. University of Chicago Press, Chicago, second edition.
- Mitchell, Tom M. 1996. *Machine Learning*. McGraw Hill, New York, US.
- Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka et al., editor, *Approaches to Natural Language*. Reidel, Dordrecht, pages 221–242.
- Moran, Douglas B. 1988. Quantifier scoping in the SRI core language engine. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics (ACL'88)*, pages 33–40.
- Moran, Douglas B. and Fernando C. N. Pereira. 1992. Quantifier scoping. In Hiyun Alshawi, editor, *The Core Language Engine*. MIT Press, Cambridge, MA, pages 149–172.
- Nerbonne, John. 1993. A feature-based syntax/semantics interface. In A. Manaster-Ramer and W. Zadrozny, editors, *Annals of Mathematics and Artificial Intelligence (special issue on mathematics of language)*. pages 107–132. Also published as DFKI Research Report RR-92-42.
- Park, Jong C. 1995. Quantifier scope and constituency. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL'95)*.
- Pedersen, Ted. 2000. A simple approach to building ensembles of naïve Bayesian classifiers for word sense disambiguation. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, pages 63–69.
- Pereira, Fernando. 1990. Categorical semantics and scoping. *Computational Linguistics*, 16(1):1–10.
- Pollard, Carl. 1989. The syntax-semantics interface in a unification-based phrase structure grammar. In S. Busemann, C. Hauenschild, and C. Umbach, editors, *Views of the Syntax/Semantics Interface*. pages 167–185. KIT-FAST report, number 74.
- Pollard, Carl and Eun Jung Yoo. 1998. A unified theory of scope for quantifiers and WH-phrases. *Journal of Linguistics*, 34(2):415–446.

- Ratnaparkhi, Adwait. 1997. A simple introduction to maximum entropy models for natural language processing. Technical Report 97-08, Institute for Research in Cognitive Science, University of Pennsylvania.
- Santorini, Beatrice. 1990. Part-of-speech tagging guidelines for the Penn Treebank project. Technical Report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.
- Soon, Wee Meng, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.
- van Halteren, Hans, Jakub Zavrel, and Walter Daelemans. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2):199–229.
- VanLehn, Kurt A. 1978. Determining the scope of English quantifiers. Technical Report AITR-483, Massachusetts Institute of Technology Artificial Intelligence Laboratory, Cambridge, MA.
- Woods, William A. 1986. Semantics and quantification in natural language question answering. In B. J. Grosz, K. Sparck Jones, and B. L. Webber, editors, *Natural Language Processing*. Kaufmann, Los Altos, CA, pages 205–248.