Treebank Parsing and Knowledge of Language

Sandiway Fong, Igor Malioutov, Beracah Yankama, and Robert C. Berwick

Abstract Over the past 15 years, there has great success in using linguistically annotated sentence collections, such as the Penn Treebank (PTB), to construct statistically based parsers. This success leads naturally to the question of the extent to which such systems acquire full "knowledge of language" in a conventional linguistic sense. This chapter addresses this question. It assesses the knowledge attained by several current statistically-trained parsers in the area of tense marking, questions, English passives, and the acquisition of "unnatural" language constructions, extending previous results that boosting training data via targeted examples can, in certain cases, improve performance, but also indicating that such systems may be too powerful, in the sense that they can learn "unnatural" language patterns. Going beyond this, this chapter advances a general approach to incorporate linguistic knowledge by means of "linguistic regularization" to canonicalize predicate-argument structure, and so improve statistical training and parser performance.

1 Introduction: Treebank Parsing and Knowledge of Language

Parsers statistically trained on corpora like the Wall Street Journal/Penn Tree Bank have steadily improved their performance. However, despite these gains, it is well-known that such systems often perform poorly on novel sentences

S. Fong (✉)
University of Arizona, Tuscon, AZ 85721, USA
e-mail: sandiway@email.arizona.edu

I. Malioutov · B. Yankama · R.C. Berwick
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
e-mail: igorm@mit.edu; beracah@.mit.edu; berwick@csail.mit.edu

A. Villavicencio et al. (eds.), *Cognitive Aspects of Computational Language Acquisition*, Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-31863-4_6,

outside their training datasets, due to the sparsity effects that reflect the "long-tail" Zipf-distributional rarity of linguistic constructions and head-dependency relations (see Collins [15], among many others). Klein and Manning [27] summarize the situation in this way:

> As a speech person would say, one million words of training data just isn't enough. Even for topics central to the treebank's WSJ text, such as stocks, many very plausible dependencies occur only once, for example, *stocks stabilized*, while many others occur not at all, for example, *stocks skyrocketed*.

Our experiments below suggest that sufficiently complex linguistic constructions exhibiting non-local dependencies may often pose problems for a parsing model that takes a static view of syntactic structure – a model unable to systematically relate the passive form of a sentence to its active counterpart, or a declarative sentence to a corresponding derived interrogative. While often effective, simply adding more data should not be invariably seen as a substitute for incorporating explicit linguistic constraints into parsing models. Indeed, the successful use of an alternative model of syntactic structure, Combinatory Categorial Grammar (CCG), as implemented in several recent systems such as the C&C parser [11] and by Hockenmaier [22, 23] may be seen as a concrete demonstration that sometimes the representation of syntactic knowledge, rather than data sparsity, plays a more important role in parser performance.

Moreover, as evidenced by the Penn Treebank, more challenging linguistic mechanisms may have the least amount of data available for learning. The problem is only exacerbated if we examine resource-impoverished languages. Language acquisition is a classic instance of a scenario where adding more data is not one of the available options for resolving the data sparsity problem. A viable computational treatment requires model-level changes to address this issue.[1]

In fact, our experiments below indicate that statistical parsing stands to benefit from a much more restrictive learning regime that inherits insights from language acquisition. On this view, parsing models should be judged based on their ability to recover and discriminate between different types of syntactic mechanisms rather than on incremental improvements from adding training data to alleviate the data sparsity problem. Similarly, the ability of a model to learn an unnatural syntactic mechanism detracts from its ability to discriminate between syntactic constraints observable in human language. Conversely, insights from our experiments can be

[1] We note that there have been recent proposals that suggest that "linguistic mastery does not need to be available early in the course of language development" and that "the acquisition of usage-based and fixed-form patterns can account for . . . [the] syntactic burst [occuring around age two to three]" [39]. It is uncontroversial that some fixed form patterns are memorized by children, and equally that complete linguistic mastery of syntax is delayed until the age of eight or later, as first established by the work of Carol Chomsky [10]. However, while it "need not" be "available early", in point of fact, empirically, it has long been established that 'telegraphic speech' is not indicative of the full scope of syntactic comprehension at the ages of 2–3; rather, many aspects of syntax are acquired by this age, but telegraphic speech does not reveal these abilities and reveals processing difficulties such as memory limitations [20, 47].

brought to bear on approaches to language acquisition. Syntactic mechanisms might be more effectively acquired and discriminated if they are characterized in terms of canonical argument analysis.

More generally, in this chapter we will focus on an assessment of gaps in the "knowledge of language" acquired by statistically-trained parsers, attempting to sort out which of these might arise from limited training data and lead to parameter estimation problems with associated parsing models, and which might arise from underlying grammatical frameworks and benefit from the insights of linguistic theory.

We note that often the two sources of error are not complementary. Adding more data relevant to a particular syntactic construction may resolve parsing mistakes, but at the same time it may be symptomatic of a systematic problem with the model. When asked to chose between two solutions, their relative ability to scale up and generalize to new instances is the critical consideration. For example, a model that needs a passive form for each active counterpart observed in the data to be able to parse the passive variant should be less preferred to a model that explicitly models the passive and is able to analyze and generate such a form automatically. This is the basic conclusion we draw from our analysis of passive sentences, and it is not simply a question about data sparsity.

We should emphasize at the outset that we have probed questions like these by constructing entirely new experiments, not simply covering familiar ground about the ever-present issue of data sparsity in statistical parsing. To the best of our knowledge, all our experiments and their results are new. The analysis of passive errors and the method we apply to canonicalize argument structure to improve passive parsing performance is also novel, as far as we have been able to determine. Similarly, our analysis of wh-questions does not simply rehash the approach of Rimmell et al. [44]. Finally, our application of an "unnatural" language learning litmus tests, while drawn from the psycholinguistic literature as in [36], has not been extended to current statistical parsers. In all of these situations, our ultimate goal is to seek ways of improving parsers by determining whether such systems have typical failure modes that can be discovered, as well as whether these failures need to be remedied.

To begin, such an assessment of "knowledge of language" poses a real challenge. Parsers are typically designed from the start to solve a very particular engineering task that is quite different from the way that a linguist might assess knowledge of language. Roughly speaking, statistically-based parsers learn how to select a "most likely" analysis with respect to all the parses they have been trained on and all the parses they can generate. They only choose among possible parses, standardly using either generative or discriminative estimation methods. In this sense, they do not directly adjudicate among "grammatical" and "ungrammatical" sentences.[2] Such a

[2]As noted in [41] and [48], despite the fact that statistically-based parsers have used both sorts of estimation methods, the underlying statistical models for both generative approaches as well as discriminative approaches using what are called "latent variables" – probabilistic and weighted context-free grammars, respectively – turn out to be equivalent in their expressive power.

probabilistic "remembrance of parses past" is not the same as the replicability of linguistic knowledge conventionally probed by grammaticality judgements.

Indeed, it is not immediately obvious how to align grammaticality judgements with probabilities. There is no agreed-upon unification. While some authors, e.g., Abney [1] maintain that the grammaticality-probability distinction should be kept firmly apart, still others argue differently, e.g., [29], p. 33:

> The parser that an ML [machine learning] system produces can be engineered as a classifier to distinguish grammatical and ungrammatical strings.

While a more detailed consideration of this point lies beyond the scope of this chapter, it suffices to observe that, as noted in [12], one cannot simply provide a probability threshold, ϵ, such that for all probability values greater than ϵ, a parse is grammatical, otherwise ungrammatical. In this case there could be at most $1/\epsilon$ grammatical sentences, and the corresponding language would be finite. Observe that the standard assumption for probabilistic context-free grammars assumes an exponential distribution of probability mass with respect to generated sentence length, so that sentences longer than a certain length have vanishingly small probability mass. ~~Thus as noted in the main text, such~~ a language is effectively finite. If anything, to the extent that such parsers are intended to model an actual corpus, they presumably reflect actual language *use*, (in the case of the PTB, newspaper writing), and so a complex mix of syntactic, lexical-semantic, world/encyclopedic knowledge, processing load, and other similar factors. This is not coextensive with the conventionally abstract, linguistic notion of linguistic *competence*, that deliberately idealizes away from this mix, though there are familiar points of contact.

Consequently, in this chapter we will typically base our assessments simply on what parsing systems can and cannot do well. To consider an introductory example of the assessment methods we will use, even in simple cases many corpus-trained parsing systems cannot recover correct verb argument structure. Consider a passive construction such as that in Ex. 1 below:

~~Mary~~ was kissed by the guy with a telescope on the lips.

Many (perhaps most) parsers trained on the PTB will tend to attach the Prepositional Phrase (PP) *on the lips* incorrectly to the PP *a telescope* because most of their training data follow such a form. In contrast, the corresponding active form, Ex. ~~1~~ below, is easily parsed correctly by such systems, because the Subject NP-PP combination is no longer located near the ambiguous PP attachment point:

~~The~~ guy with a telescope kissed Mary on the lips.

Such examples are not just hypothetical. For instance, Fig. 1 shows that sentence #404 of section 23 of the PTB, *Measuring cups may soon be replaced by tablespoons in the laundry room*, is parsed incorrectly exactly in this way by two state-of-the-art parsers, the Stanford unlexicalized context-free parser [27] and Bikel's re-implementation of the Collins parser [4]. In all these cases, the PP *in the laundry room* is incorrectly attached as a modifier of the object NP *tablespoons*.

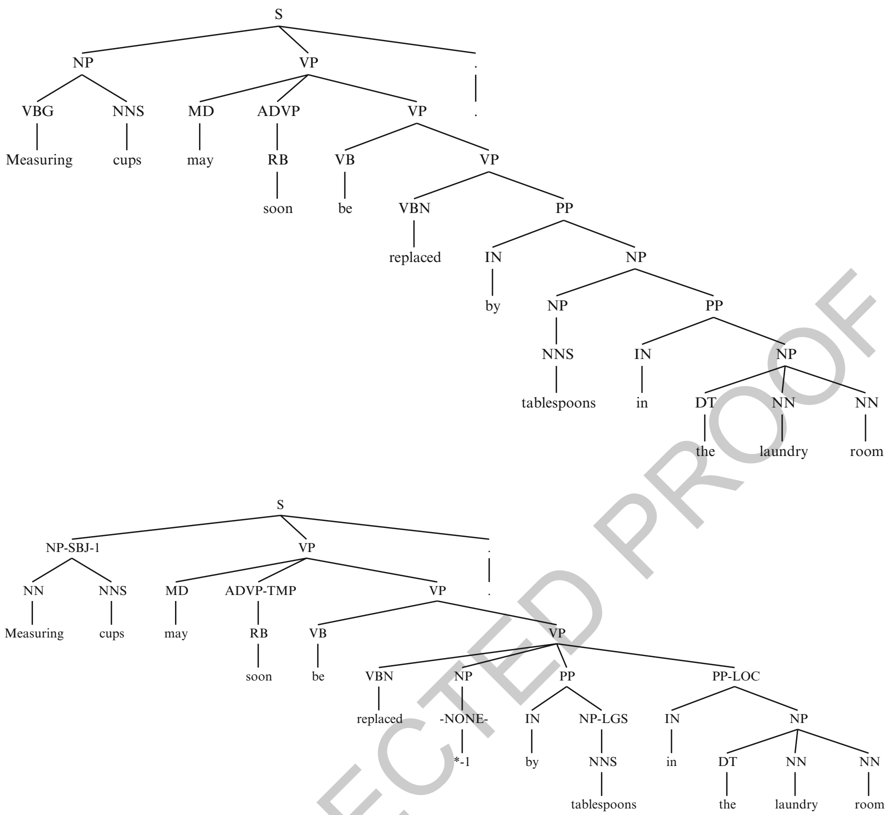


Fig. 1 The Bikel/Collins and Stanford unlexicalized parsers both mis-analyze sentence number 404 in section 23 of the PTB. The *top* half of the figure shows the result of parsing using either Bikel's reimplementation of the Collins parser or the Stanford unlexicalized parser. The *bottom* half of the figure shows the corresponding "gold standard" PTB structure

~~As in~~ the remainder of this chapter, with some exceptions we will typically test examples on a range of probabilistic parsers in an attempt to avoid the idiosyncrasies of any particular implementation and achieve some measure of robustness in our test results. In this case, in addition to the two parsers illustrated in the main text, the Berkeley parser [40] and the C&C combinatory categorial grammar parser [18] both output the same, incorrect attachment. The Malt dependency parser version 1.4.1 [37] also outputs an incorrect dependency between *in* and *tablespoons*. In contrast, both the "factored" Stanford lexicalized-dependency parser [28] and the Charniak-Johnson parser [6] *do* output the correct attachment.

Examples such as these suggest that verb argument structure might be more easily recoverable when sentence structure is represented in some canonical format that more transparently encodes grammatical relations such as Subject and Object. In other words, if the arguments of predicates are in a fixed syntactic position in training examples, then we might expect that this regularity would be simpler for a statistically-based system to detect and acquire. More generally, it has often been observed that what makes natural languages difficult to acquire or parse is that phrases are displaced from their canonical positions, not only in passives, but in topicalization, wh-movement, and many similar constructions. Each of these constructions breaks the transparent link between predicates and arguments. In Sect. 5 below, we shall see that one can remedy at least some of these difficulties by adopting a representation that is arguably closer to the one that certain linguistic theories assume, where the argument of the main verb has been ‘replaced’ in its canonical Object position, as in Ex. 1. There are other representations one might adopt to handle this particular problem, for example, a combinatorial categorial grammar (CCG) that explicitly relates displaced phrases to their “gaps.” As we noted earlier, this does not necessarily ensure success.

Following the lead of this illustrative example, in the remainder of this chapter we will focus on the following selection of challenging areas for parsers trained on corpuses like the PTB:

1. **Wh-questions.** As has often been noted, the PTB corpus contains a very small number of questions – unsurprisingly, since it consists of *Wall Street Journal* newspaper articles [34]. Out of the 39,822 sentences in the standard training sections 02–21, there are only 128 “root” level questions, such as training data sentence #85, *What's next?* and four other similar questions. More than 70 % of these are Subject wh-questions There are 61 additional wh-questions that appear in embedded quotational contexts, e.g., *“What's he doing ” , hissed my companion*, and 96 root level auxiliary inverted questions, e.g., *Was this why some of the audience departed before or during the second half* . In short, by all measures, the training data for wh-constructions and questions is exceptionally sparse. Moreover, the statistically-trained parsers we examine in this chapter do not receive data in the form of “more ill-formed” examples that differ, say, by just a single word in a different order, such as, *Who asked who bought what* vs. *Who asked what who bought*. These systems must therefore learn such nuances from just one or two positive examples.
2. **Tense marking.** Tense is a good example of a linguistic phenomenon that, like displacement in wh-questions, may be “spread out” over several, not necessarily adjacent words. For example, in an English yes-no question, tense must be realized overtly at the front, while the corresponding main verb need not have an overt morphological indicator of tense: thus we have the PTB example, *Do you think the British know something we don't* , where *do* carries tense and *think* does not. We will investigate whether statistically-trained systems can “capture” part of the English tense system by examining examples of verbs that are ambiguously marked for tense, such as *read* or *cost*.
3. **Passives.** As noted in our introductory example, the placement of a verb's argument in Subject position, along with the possibility of an Agentive “by” phrase can lead to parsing difficulties.

4. **"Unnatural" language constructions.** Finally, while the previous topics all examine a particular parsing task – essentially, structural language patterns – that one would like a trained parser to detect easily, there are also non-attested language patterns that trained parsers should be able to detect only with great *difficulty*. A cognitive-faithful parser should have the same problems acquiring "unnatural" language patterns as people do. But what do mean by unnatural? By this we do not mean patterns that are challenging for people due to processing constraints, e.g., the classic examples of center-embedded or garden path constructions. Rather, what we will mean by "unnatural" language constructions are examples of the sort studied in some detail by Musso et al. [36] via artificial grammar learning and fMRI experiments. They covered two sorts of unnatural rules: (1) "counting" rules, that is, linguistic rules that, say, could form the negation of a declarative sentence by inserting a special word at a particular point in a sentence, say, always immediately after the third word; (2) "mirror image" rules, that is, linguistic rules that, say, could form the interrogative of a declarative sentence by inverting the word order of the declarative sentence, saying it in reverse. In their study, [36] constructed a set of unnatural rules, unattested in any natural language. Here is their description of the second "unnatural" rule, which is the one in Sect. 6 that we will attempt to reproduce as closely as possible in our experiments with statistical parsers, from [36], p. 775:

   > The second rule required that the interrogative construction be built by inverting the linear sequence of words of a sentence. For example, "*I* [1] *bambini* [2] *amano* [3] *il* [4] *gelato* [5] or "The children love ice-cream" becomes *Gelato* [5] *il* [4] *amano* [3] *bambini* [2] *il* [1].

   Musso et al. found that people had great difficulty mastering artificial rule systems of this sort. If they were learned at all, they were learned, as if they were non-linguistic 'puzzles,' activating very different brain regions than those lit up during normal language rule processing. Smith et al. [49] reported a similar finding, again using an artificial grammar learning paradigm. Here it was discovered that an autistic linguistic "savant" could not learn "unnatural" grammatical rules. In contrast, while adults could learn these rules, but again, only with great difficulty. In a related area, others (e.g., [33]) have noted that the same issue arises with respect to artificial neural network learning in the paradigm case of English past tense over-regularization. Neural network systems that are constructed to report the probability of the next word or form in a sequence are apparently "unnatural" to the extent that they can learn sentence reversals just as easily as normally ordered word sequences. Note that this is a case where the neural network simulations do equate "grammaticality" with "likelihood." What all these results come to is the same: we do not want a "natural" learning system to be *too* flexible, having capacities beyond those found in people.[3]

[3] See, e.g., [9] and [2] for additional discussion of the lack of non-counting and palindromic rules in natural language, including syntax and phonology. It is known in certain sociological settings

2 Experimental Methods

We carried out our experiments on as broad a range of publicly available statistically-trained parsers as possible, subject to the broad constraint they all could be trained on the same, standard subsections of the *Wall Street Journal* version of the Penn Tree Bank III. In this we strove to follow the same procedure and roughly the same coverage as in the comparative study carried out in [13], p. 51:

> Constituent parsers and dependency parsers all have the appropriate level of sophistication, but a wide variety of different grammars and conceptual frameworks that makes comparing them difficult. However, there is one class of parsers that is both numerous and up-to-date, and covers a variety of different algorithms which all use the same output format (bar a few small details). These are sometimes referred to as treebank parsers as they are usually trained and optimized on the PTB and produce output conformant with its standards.

2.1 Parsing Systems Used

AQ2

The systems that were used for the experiments are given in Table 1. Not all of these systems could be used for all experiments, due to certain resource requirements. Such details will be noted in what follows. Among the publicly available systems, we selected the most extensively cited and most widely used parsers. We cannot hope to exhaust the full range of parsers now publicly available, particularly dependency parsers. For example, we could not include the Melamed/Turian discriminative parser [52]. We leave such extensions for future research. Additional details about the grammatical models and the training/testing procedures used will be covered as they arise.

2.2 Training Data, Testing, and Evaluation

In order to ensure that results would be as comparable as possible, we retrained most of the parsers on sections 02–21 of the PTB III, even when they came with "pre-built" estimated models on this training data (as with the C-J, Berkeley, and Stanford parsers).[4] Due to limited access to the original materials and other computational constraints, we were not able retrain the CJ-R parser. As a result, in what follows we

---

that palindromic forms are used, e.g., the Australian butchers' market language. But all indications here are that this such behavior remains "puzzle based."

[4]We attempted to use training settings that matched those for the parsers' "pre-built" models as far possible. For example, we used the settings provided in the Stanford parser directory under `makeSerialized.csh` for the so-called `wsjPCFG` model. In the case of the BC-M2 parser, we used the settings given by `collins.properties` since we wanted to ensure replicability with standard results.

Table 1 The treebank parsers chosen for this investigation

Parser	Abbreviation	Release used	Citation
Bikel-Collins Model 2	BC-M2	1.2 Oct 08[a]	[4]
Berkeley "coarse to fine"	Berkeley	1.1, Sept 09[b]	[40]
Stanford unlexicalized	Stanford-unlex	1.6.3[c]	[27]
Stanford factored dependency	Stanford-fact	1.6.3[c]	[28]
Charniak "coarse-to-fine"	CJ-I	Nov 09[d]	[5]
Charniak-Johnson reranking	CJ-R	Nov 09[d]	[6]

[a] http://www.cis.upenn.edu/~dbikel/download/dbparser/1.2/install.sh
[b] http://code.google.com/p/berkeleyparser/downloads/detail?name=berkeleyParser.jar
[c] http://nlp.stanford.edu/software/stanford-parser-2010-07-09.tgz
[d] http://web.science.mq.edu.au/~mjohnson/code/reranking-parser-Nov2009.tgz

used only the CJ-R pre-built model. In addition to using this standard training data, we carried out various experimental manipulations followed by data augmentation and retraining that will be described in later sections. For evaluation we used the standardly available `evalb` package [46].

3 Case Study: Parsing Wh-Questions and QuestionBank

We first return to the area of wh-questions outlined briefly in Sect. 1. For the purposes of this chapter, we will put to one side the question of how to link wh-words and phrase such as *what* or *which problem* to their 'gaps', for example, the link between *what* and the object position after *buy* in a sentence such as *What did John buy*. While this is an important topic, full analysis of this problem is beyond the scope of the current chapter; see [44] and [18] for combinatory categorial grammar approaches that address this issue. Instead we will focus solely on the question of how well correct ~~sentence is~~ recovered.

Why would parsing problems arise even if we put this issue aside? The reason is that in the standard training sections of the PTB, wh-phrases are most often used as relative clauses, not as questions (in a ratio of approximately 10,000:1). It would not be surprising, then, if a true wh-question was parsed as if it were a relative clause. Using standard PTB notation, we would then expect wh-questions parsed incorrectly as an S embedded within an SBAR, rather than, correctly, as an SQ (a sentential question) embedded within an SBARQ. (See Fig. 2 below for a representative example of this distinction.)

To be concrete, a conventional linguistic assessment about knowledge regarding wh-questions often begins with a "graded" list of examples such as those in Ex. 3 below, where the first sentence is an "echo question." This is followed by a semantically similar wh-interrogative sentence. The next three examples are then listed in roughly an order of descending acceptability to native English speakers (hence the asterisks placed before them).

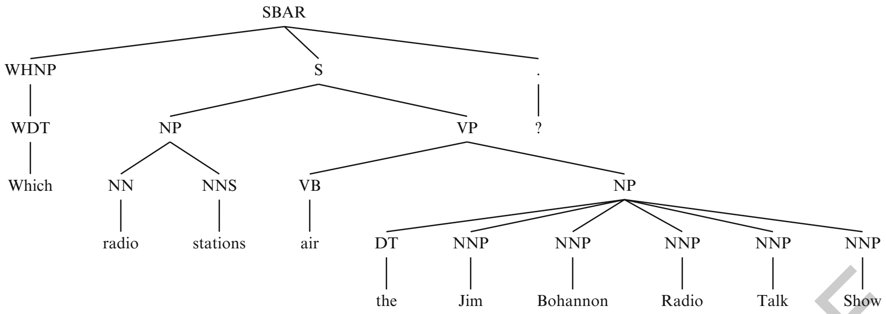


Fig. 2 An example of a wh-question parsing error for the sentence, *Which radio stations air the Jim Bohannon Radio Talk Show?* This is the output from the BC-M2 parser

~~a. Bill will solve which problem?~~
b. Which problem will Bill solve?
c. Which problem Bill will solve?
d. Bill solve which will problem?
e. Which problem Bill solve will?

How might we use such examples to test the linguistic knowledge acquired by a statistically-trained parser? Note that even if a sentence is "ill-formed" like the last three above, then a probabilistic parser will still try to do the best it can, and return the most likely analysis, even a partial or incorrect one, with respect to the parsed examples it has already been trained on. That is in some respects an appropriate response to what such systems have been designed to do, one means to add robustness. As we described in the introduction, this might be a perfectly valid way to proceed from an engineering standpoint; factoring in gradience judgements of this sort remains an area to explore that lies beyond the scope of the present chapter. Further, while we might expect that the probability scores returned by the parser for the last three sentences could be worse than those for the first two, likelihood scores would probably vary anyway given slightly different local contexts and the successive history of various local rule choices set against what has been seen in the training corpus. In addition, if a parser is "lexicalized" then the actual word information (e.g., whether the verb is *solve* or *try*) is typically propagated to the head of a phrase (in this case, the Verb Phrase (VP)), and in this way specific lexical items may play a role in influencing what analysis path is taken.

Putting this question of assessing grammaticality to one side, we therefore focus instead only on the problem of producing the correct parse, rather than any likelihood score that denotes relative acceptability or grammaticality. That this is a real problem may be seen in Fig. 2 below, which displays an incorrect parse of a wh-question sentence produced by the BC-M2 parser, on an example sentence taken from an actual corpus of wh-questions, QuestionBank, that we describe immediately below.

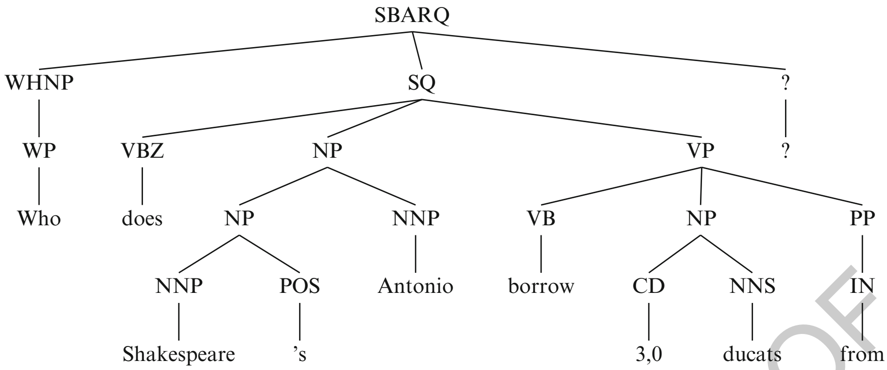


Fig. 3 Parse structure assigned to the "Who does Shakespeare..." sentence by the downloaded QuestionBank used in the current analysis

3.1 Augmenting the Training Data

There have been several approaches to remedying this problem by adding additional wh-question training sentences. In particular, Judge et al. [26], Rimmell et al. [44], and Nivre et al. [38] have built systematic "unbounded dependency" question treebanks.

We did not have access to these last resources, so we drew instead on a recently-built publicly accessible 4,000 sentence database, QuestionBank, constructed by Judge et al. [26]. This is a curated database of 2,000 questions drawn from the TREC question-answering (QA) domain and 2,000 questions from the Cognitive Computation Group at UUIC.[5] A representative example from this version of the QuestionBank is, *Who does Shakespeare's Antonio borrow 3,0 ducats from?*, as displayed in Fig. 3. Note that unlike the PTB II/III, this downloaded version did not contain information about the location of the underlying argument positions of displaced phrases, e.g., that *Who* serves as the object argument *from*) in the preceding example. From our perspective this was satisfactory because, unlike the research reported on in [26, 44], or [38], we were interested solely in the question of whether statistical parsers could learn correct structural analyses.

Note that while QuestionBank represents approximately a 10 % addition to the number of sentences to the baseline training set, most of these wh-question sentences are typically far shorter than those in the PTB II, with a median sentence length of ten words – unsurprising since these are questions culled from a question-answering domain as opposed to the written *Wall Street Journal* newspaper article domain.

[5] The full database was obtained by download from http://www.computing.dcu.ie/~jjudge/qtreebank/. A handful of errors in corpus annotation were corrected in this downloaded dataset.

Table 2 Labeled precision, labeled recall, and F-Scores for baseline and wh-trained parsers, using question training/test data from QuestionBank (QB). The last column displays F-scores for these parsers' performance on only the standard baseline section 23 of the WSJ

Parser type	Labeled precision, %	Labeled recall, %	F-score, %	F-score, % WSJ Sect. 23
BC-M2 baseline	80.87	71.25	75.76	85.63
BC-M2+QB	91.08	~~81.7~~	86.18	85.79
% improvement	12.63	14.67	13.75	
Stanford-unlex baseline	66.26	69.32	67.57	85.54
Stanford-unlex+QB	81.72	80.92	81.32	85.55
% improvement	22.33	22.01	20.03	
Stanford-fact baseline	~~62.5~~	65.57	64.00	88.71
Stanford-fact baseline + QB	88.71	87.41	88.06	88.59
% improvement	20.53	15.60	17.99	
CJ-I baseline	84.65	71.81	~~77.7~~	86.55
CJ-I+ QB	90.31	80.65	85.21	88.13
% improvement	6.69	12.31	9.67	

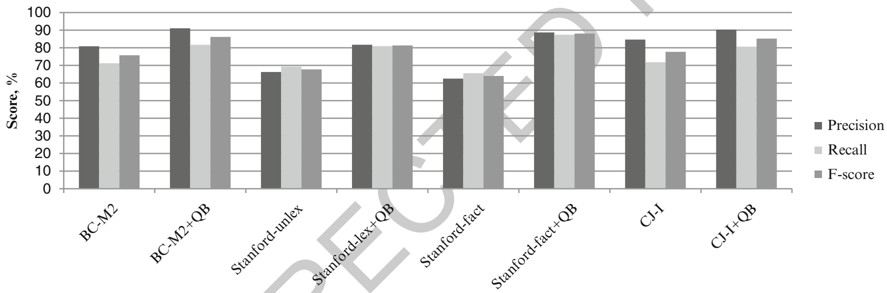
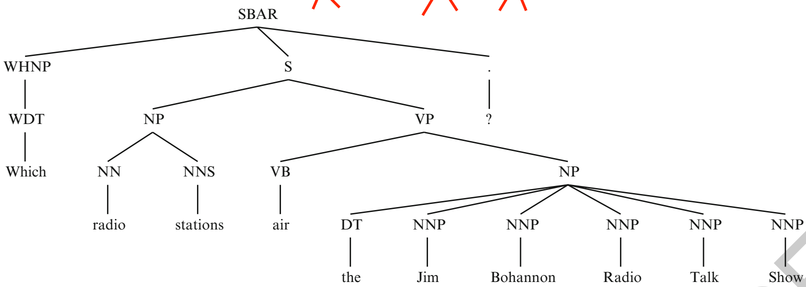


Fig. 4 Labeled precision, labeled recall, and F-scores for the parsers trained and tested on the QuestionBank corpus, both before and after training on QuestionBank

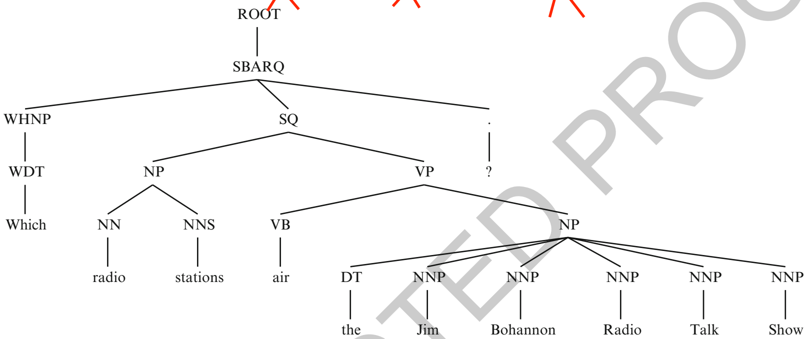
We divided the 4,000 QuestionBank sentences into an 80 % training portion and a 20 % testing portion. We tested four parsers: BC-M2; Stanford-lex; Stanford-fact; and CJ-I. We tested each of these four parsers on two training-test sets: (1) the baseline conventional PTB training set; (2) the 80 % Question Bank sample, eight experiments in all.

Table 2 gives the complete numerical results of these eight runs, while Fig. 4 displays the results visually, as histograms of the precision, recall, and F-score before/after performance. Both reveal a substantial improvement across all parsers. For example, Stanford-unlex parser had labeled precision/labeled recall scores of 66.26 %/69.32 % before training, and 81.72 %/80.92 % after training, a considerable gain of 15 and 10 % points, respectively (a 20.53 % and 15.60 % increase). The CJ-I parser's scores were boosted from 84.65 %/71.81 % to 90.31 %/80.65 % This was the smallest percentage improvement, due probably to the fact that even before wh-training the CJ-I parser already performed quite well. Still, increases with

Parse(i)–baselineBC-M2parse



Parse(ii)–baselineStanford-unlexparse



Parse(iii)–parseafterwh-training(bothparsers)

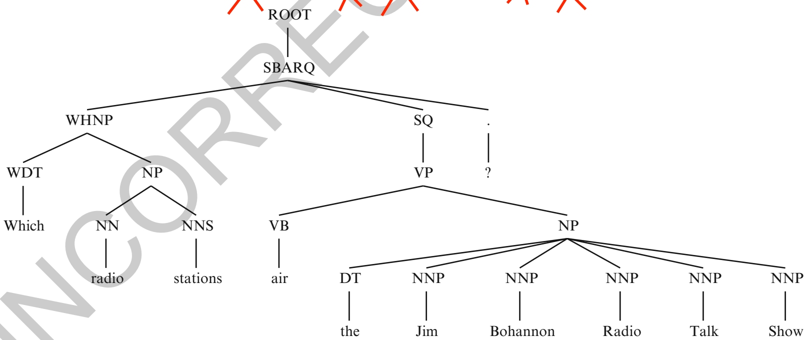


Fig. 5 An example of wh-parsing improvement after wh-training for the test sentence *Which radio stations air the Jim Bohannon Radio Talk Show?* The topmost portion (i) shows the BC-M2 parse before training, with an erroneous S node at the *top*, and the WHNP and NP as distinct trees. Similarly, the Stanford-unlex parse incorrectly separates the WHNP and the NP, while getting the SQ node correct, *middle* display (ii). The *bottom* portion (iii) exhibits the correct parse output by both the BC-M2 parser and the Stanford-unlex and Stanford-fact parsers after wh-training on QuestionBank

wh-training were quite substantial at 6.69 % and 12.31 %, with an overall F-score increase of 9.67 %. Importantly, as the last three columns of the table show, this improvement did not come at any apparent cost in precision/recall for the standard WSJ section 23. For example, the Stanford-unlex parser after additional wh-training got an F-score 85.55 %, on WSJ section 23, as compared to a baseline F-score of 85.54 %. In most cases, the additional wh-examples improve performance.

A representative example of a parse that is greatly improved by wh-training is depicted in Fig. 5, for the test data sentence, *Which radio stations air the Jim Bohannon Radio Talk Show?* Before wh-training, none of the parsers could correctly analyze this sentence. For instance, as expected, the Bikel-Collins parser mis-analyzes the words *which radio stations* as an S dominated by an SBAR, and also mis-parses *which radio stations* as distinct *WHNP* and *NP* phrases (part (i) of the figure). The Stanford-unlex parser does better, without any wh-training; it parses the sentence correctly as an SBAR dominating an SQ. However, it also fails to combine *which radio station* into a single wh-phrase (see (ii) in the figure). After training, both parsers produce 100 % gold-standard parses, shown at the bottom of Fig. 5, panel (iii).

We conclude that the 3,200 questions in QuestionBank, provide a substantial performance boost to wh-question parsing, enough to overcome any deficiencies in the original PTB. However, we note that this puts to one side the question of linking wh-elements with their "underlying" argument structure, as noted by Rimmell et al. [44], among others. In this sense, the fundamental representational question is still not addressed.

4 Parsing and Tense: The Case of *Read*

In a Linguistic Society of America pamphlet, Ray Jackendoff [24] considered a "text reading" puzzle as an example of what is impossible for a computer to accomplish without knowledge of language: in particular, the task of determining the pronunciation of the orthographic form *read*, which can be pronounced as *red* or *reed* depending on context. The sentences considered by Jackendoff are reproduced in (4); we will consider additional examples as well. In these examples, [24] introduced *will* as a deliberate complication since it can be either a Noun or Modal verb. Apparently, this was to illustrate that simply looking at adjacent words, without any sophistication, would be problematic. In any case, if this issue arises at all, we dealt with it by substituting *should* or *stock* for *will*, as appropriate. The results remained the same, so for our purposes this additional complication was ignored in what follows.

~~a. The~~ girls will read the paper. (*reed*)
b. The girls have read the paper. (*red*)
c. Will the girls read the paper? (*reed*)

Table 3 The Penn Treebank verbform tagset

Tag	Description	*Example*
VB	Verb, base form	*write*
VBD	Verb, past form	*wrote*
VBG	Verb, gerund or present participle	*writing*
VBN	Verb, past participle	*written*
VBP	Verb, non-3rd person singular present	*write*
VBZ	Verb, 3rd person singular present	*writes*

d. Have any men of good will read the paper? (*red*)
e. Have the executors of the will read the paper? (*red*)
f. Have the girls who will be on vacation next week read the paper yet? (*red*)
g. Please have the girls read the paper. (*reed*)
h. Have the girls read the paper? (*red*)

It should be clear from the examples in (4) that a computer program needs to possess knowledge of the English auxiliary/main verb system along with basic properties of sentence phrase structure in order to correctly carry out this task. The PTB assumes a part of speech tagset that identifies and distinguishes among different forms of a verb, as shown in Table 3. This information ought to suffice, since these values are enough to fix a deterministic decision procedure to pronounce *read* correctly. Note that such a parsing system must be able to associate, e.g., the tense marking on a word like *will* with the correct tense of the verb *read* that appears later in the sentence. General agreement phenomena such as this have been a staple of linguistic analysis for more than 60 years [8]. A related issue appears with other verb forms such as *cut* or *cost*, that are ambiguous with respect to their tense information in the third person (e.g., *they cut/they have cut*). In this case, though their pronunciation is also identical, there is still a problem in picking the right tense label for the verb, as we shall see.

One might reasonably expect a parser trained on nearly 40,000 sentences to have acquired basic English sentence structure and properties of the auxiliary and verbal system, and thus be able to decode the examples ~~in (4),~~ correctly identifying the appropriate tag for *read* in each case, thus solving the "text reading machine problem" posed by Jackendoff. This is the question we shall examine here.

For example, the structure recovered by the Berkeley parser in the case of 4(b), correctly identifying *read* as VBN, is given in Fig. 6 on the left. (In the case of *read*, only the VBD and VBN forms should be pronounced as *red*.)

However, the Berkeley parser is not always correct. The bottom part of Fig. 6 illustrates the corresponding Berkeley parse for 4(h). Here the sentence has been properly identified as an interrogative (category label SQ) but the parser nonetheless has fails to assign the correct VBN tag to *read*. (The assigned tag VB will result in a pronunciation of *reed*.)

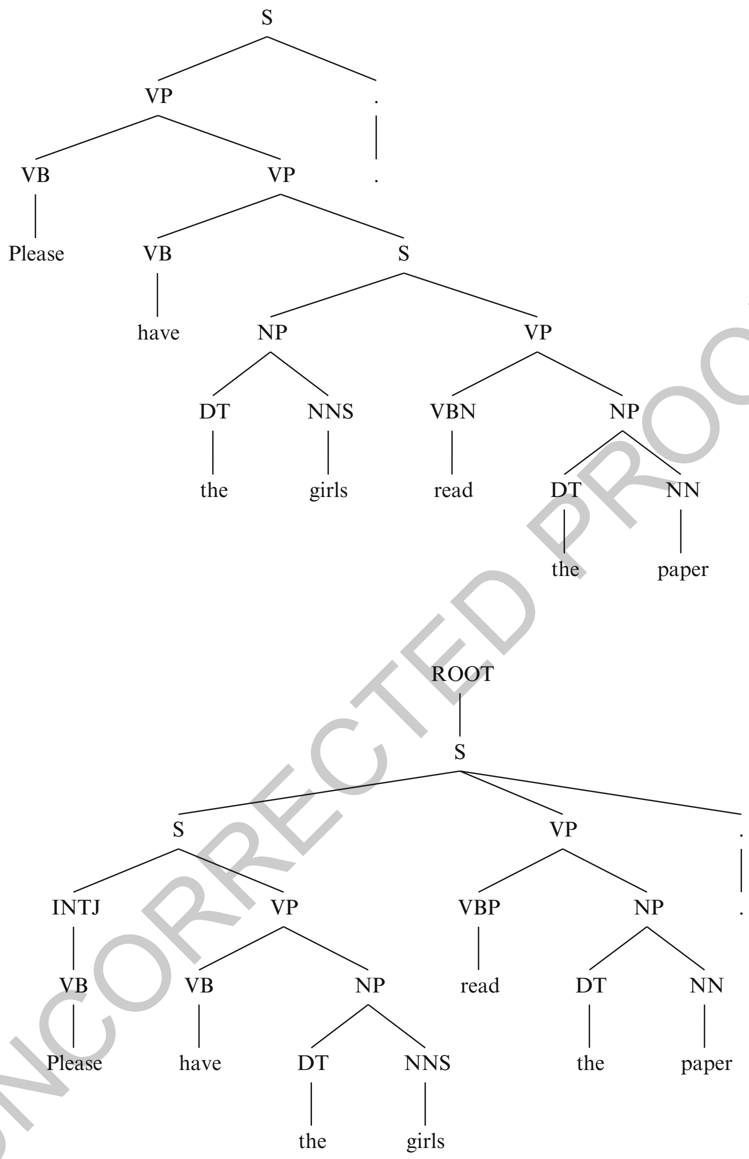


Fig. 6 Berkeley (*top*) and BC-M2 (*bottom*) parses for sentence Examples 4(b,h)

Continuing with this experiment, we examined in some detail how the Jackendoff *read* sentences are analyzed by our suite of statistically-based parsers, all trained on the same sections of the PTB. The results are summarized in Table 4. There are striking differences in performance. Even some of the output parse structures are different. (See Fig. 7 below for a display of a parsing difference with the imperative

Table 4 Parsing results for the *read* pronunciation task. All parsers trained on identical data. Incorrect outputs are flagged with an asterisk*

Example	(4a)	(4b)	(4c)	(4d)	(4e)	(4f)	(4g)	(4h)	
Correct form	VB	VBN	VB	VBN	VBN	VBN	VB	VBN	Correct
Berkeley	VB	VBN	VB	*VB	*VB	*VB	VB	*VB	4/8
BC-M2	VB	VBN	VB	*VB	*VB	VBN	*VBN	* VB	4/8
CJ-I	VB	VBN	VB	*VB	*VB	VBN	*VBN	*VB	4/8
CJ-R	VB	VBN	VB	*VB	*VB	VBN	*VBN	*VB	4/8
Stanford-unlex	VB	VBN	VB	VBN	VBN	*VB	VBP	VBN	7/8
Stanford-fact	VB	VBN	VB	VBN[a]	VBN	VBN	VBP	*VB	7/8

[a]This assumes that the parser has not misinterpreted *will* as a modal verb. The same holds for the next example

sentence Ex. 4(g).) Overall, the Berkeley parser gets 4/8 of the test sentences correct, missing 4(d–f,h).[6]

The BC-M2 parser does not have perfect performance either, with 4/8 correct, though it fails on a slightly different set of examples; it misses 4(d,e,g,h). For comparison, note that an assignment based purely on tag frequency would yield a crude baseline of 3 out of 8 correct on this task, as VB and VBN occur 45 % and 19 % of the time in the training set for *read*. It is important to observe that unlike the other parsers tested here, the BC-M2 parser ignores final sentence punctuation, so it literally cannot distinguish *Have the ...?* from *Have the ...*.

The other two lexicalized parsers, both the 'first-stage' *n*-best parser using Charniak's "coarse to fine" method and the CJ re-ranking parser, perform exactly the same as BC-M2, getting 4/8 sentences right, and missing the same sentences as BC-M2, on sentences 4(d,e,g,h).[7]

Finally, turning to the two Stanford parsers, we see greatly improved performance. If we count VBP as OK for the imperative *read* sentence, then the (simpler)

[6]As noted in Sect. 2 we tested both the Berkeley's parser's pre-built `eng_sm5` grammar, as well as our own retrained version that carried out six split-merge iterations. The results did not change. The results also remained the same when we used Berkeley parser's `-accurate` switch. In general, results did not change for any of the parsers when we substituted *stock* or *should* for *will*. Note that here the Berkeley parser is using its own part of speech tagger. If we force it to use "gold standard" part of speech tags, then it could not possibly fail in the manner we have described. However, we wanted to examine the parser's own performance, not some exogenous part of speech tagger.

[7]For CJ-I we selected the "best" (highest likelihood parse score) from the output of the CJ-I parser. In fact, in several cases, the 2nd best parse tree turned out to be the correct one; this was true, for instance, for sentence 4(h). On the other hand, just as often the best parse was correct and the 2nd best parse was incorrect, as in example 4(a). Note that the CJ-I parser serves as input to the CJ-R re-ranking parser, taking, e.g., the top-50 most likely parses and then sorting them according to a discriminative weighted feature-based scheme using features such as the degree of right-branching, or conjunct parallelism. Since the top 50 parses usually included the correct answer, the re-ranking parser at least had a chance of possibly selecting the correct answer in each case. Even so, re-ranking was ineffective, and did not change the outcome for any of the sentence examples here. See [6] for details about this re-ranking parser.

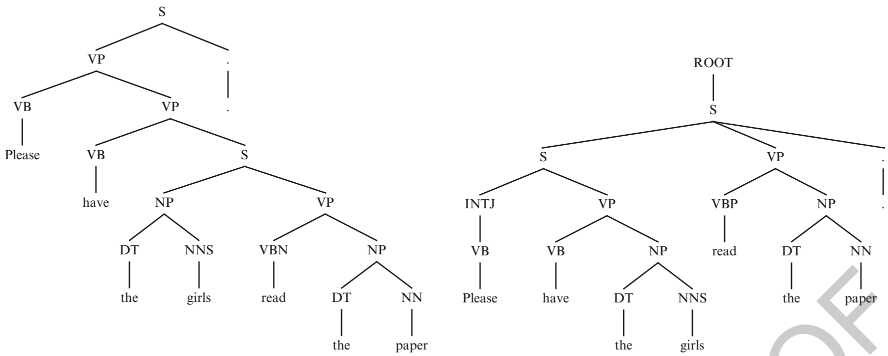


Fig. 7 Some parsers output distinct structures for the imperative *read* sentence. The *left-hand* side displays (identical) the parse output by the Berkeley, and BC-M2 parsers. (The CJ-I and CJ-R parses are also identical to this one, aside from the minor difference of labeling *have* as an AUX.) The *right-hand* side displays the output from the Stanford parsers for this same sentence

Stanford unlexicalized, probabilistic context-free parser is nearly perfect, with 7/8 sentences correct. The more sophisticated dependency-factored Stanford parser also gets 7/8 correct, (Both of these parsers also output different, arguably incorrect parses for *Please have the girls read the paper*, displaying the imperative form as shown on the right-hand side in Fig. 7.)

What accounts for the difference in the results? All of the parsers use extremely sophisticated statistical estimates, with many programming details, so it is very challenging to determine what accounts for their varying performance on particular sentences. As Bikel observes, [4], p. 188:

> With so many parameters, a lexicalized statistical parsing model seems like an intractable behemoth. However, as statisticians have long known, an excellent angle of attack for a mass of unruly data is exploratory data analysis.

We shall pursue such an exploratory path here. Let us consider first the essentially identical performance of the BC-2, CJ-I, and CJ-R parsers. As noted in [5], all these parsers are strongly "lexicalized," in the sense that they use literal word information about the heads of phrases in the linguistic sense (smoothing this if necessary by various methods). That is, instead of a rule expanding a Verb Phrase (VP) as $VP \rightarrow VNP$, these parsers modify the context-free rule to incorporate actual information about the lexical head word, e.g., the particular verb *read*. The by-now familiar advantage here is to possibly capture any special properties that distinguish *read*, from, say, *buy* – perhaps that *buy* is more frequently followed by an object Noun Phrase. Such systems thus serve as a point of contrast with the remaining parsers tested, which do not in general expand context-free rules with augmented head information. We put to one side for now the method that the factored Stanford parser uses, which is in effect to parse with both an ordinary PCFG and a lexicalized dependency model, and then combine the results by means of a joint inference model.

More specifically, we may be able to pinpoint the difficulty with the lexicalized parsers by drawing on an observation made by Charniak [5]. Charniak notes that the BC-M2 parser and the CJ-I and CJ-R parsers all make use of actual lexical information, to first "guess" whether a pre-terminal label should be, e.g., VB or VBN, p. 137:

> . . . the current parser first guesses the head's pre-terminal, then the head, and then the expansion. It turns out that usefulness of this process had already been discovered by Collins [14]. . . However, Collins . . . does not stress the decision to guess the head's pre-terminal first, and it might be lost on the casual reader. Indeed, it was lost on the present author until he went back after the fact and found it there.

While [5] notes that this method accounts for a nearly 2 % performance gain overall, there is some evidence that it also leads to precisely the observed problem with *read*, essentially one of "over-lexicalization." In particular, as explained in [3], the BC-M2 parser "guesses" the part of speech of a pre-terminal associated with *read* via a top-down generative approach, sometimes modifying the pre-terminal part of speech information. We can see the effect of this in the case of *read*. In the example *Have the executors of the will read the paper*, *read* is initially assigned the (correct) part of speech tag VBN by a pre-processor tagging step. But this is changed by the probability model's guess of the incorrect tag VB. Indeed, the same holds for the other mistakes BC-M2 makes: initially correct tags are changed to their incorrect counterparts by the parser.

Our hypothesis, then, is that the local "guessing" carried out by the generative probability model in these cases may be biased by local frequency effects in such a way as to sometimes alter the tag in the wrong direction. For example, *read* AQ3 appears in the PTB training data 29 times as a VP dominating a VB (usually with an intervening *to*, and 10 as a VBP, so in 39 contexts is pronounced *reed*. On the other head, *read* appears 24 times dominated by VBD or VBN, pronounced *red*. It is this bias that appears to be altering the results. In contrast, consider the tense-ambiguous verb *hit*, which appears 88 times as VBD/VBN and only 23 times as a VB/VBP. This distribution is the converse of *read*. Running the same sentences as in 4 through the parsers with *hit*, instead of *read*, e.g., *Have the girls who will be on vacation next week hit the paper*, we find that the number of mistakes is reduced, with the correct tag VBN replacing the incorrect VB tag in three cases. Similarly, *cost*, which has the same rough local frequency distribution as *read*, with 65 VB/VBP and 22 VBD/VBN counts, behaves as expected like *read*; so does *cut*. If this view is on the right track, then it is these local frequencies, which are sensitive to the small sampling effects of the PTB, that are at play here. Further, this same issue seems to infect the other two "lexicalized" parsers, though not to precisely the same extent: when we replace *read* with *hit*, then the CJ-I and CJ-R parsers now get sentences 4(d,e) correct (as does BC-M2), but these two parsers still fail on the last two sentences. Some kind of lexicalization effect is operating, but it is not exactly the same as that with BC-M2, perhaps because the CJ parsers augment the standard PTB part of speech categories with the addition of AUX for *have*.

Additional confirmation of the effect of lexicalization comes from examining the behavior of the unlexicalized parser, Stanford-unlex. It does not make any assumptions about lexical heads, and so we would not expect it to be subject to the variation we see with the lexicalized parsers. In fact, as shown in Table 4, it is much more successful, making only one mistake, labeling *read* as a VB in *Have the girls who will be on vacation next week read the paper yet*. Note that the addition of a lexicalized component that is grounded on dependencies, the factored Stanford model that uses both word dependencies and the Stanford unlexicalized parser to jointly infer structure, also makes a single error, but it is not the same one. Instead, it makes an error on the last *read* sentence, taking it as a VB rather than a past-tense VBD. While the reasons for these singleton errors remain obscure, it is clear that this approach works better than straight lexicalization.

It remains to account for the behavior of the Berkeley parser. While it is not lexicalized, it works by refining categories and rules by successive state-splitting. It may be that its "window size" for learning context is too narrow. The trainer uses a context window based on horizontal (h) and vertical (v) "markovization," that is, how many past horizontal ancestors are remembered, and how many vertical (parent, grandparent) ancestors are remembered, as a context for future parsing decisions. By default, these values are set to 0 and 1, respectively – that is, a context that remembers only the immediate parent node above a current position. Note that in an imperative form like 4(g), the "distance" between the verb *have* and *read* lies outside this window. In [27], larger values for h and v are systematically explored, with some evidence provided that h and v values larger than 0 or 1 may be needed for generally effective performance. It remains to explicitly test this hypothesis precisely within the context of the *read* example.

How can we improve the performance of the parsers on the *read* examples? If the effect is due to sparsity and lexicalization, then as with the wh-question case, more data might prove helpful. Here the models distributed with the Stanford parser themselves indicate that additional data of the right kind indeed can be a benefit. Along with models trained solely on the PTB, Stanford-unlex and Stanford-fact come with models trained on a selection of biological abstracts from the GENIA corpus [51], plus 96 "additional" hand-built parse trees; these are called `englishPCFG` and `englishFactored`. Importantly, the 96 "additional" hand-labeled examples include examples that are directly comparable with the *read* examples, including 11 relatively short subject questions, SQs typically with subject-auxiliary verb inversion, such as *Is what she said untrue*; and 25 wh-questions, or SBARQs, such as *Where was the fox*.[8]

Probing a bit further, if we run the *read* examples using the Stanford models based on this augmented corpus then they do perfectly, so it would seem worthwhile

[8]The remaining examples are some simple S's and a few newswire stories. The authors would like to thank C. Manning for generously sharing these additional examples with us.

Table 5 Parsing results for the *read* pronunciation task when rerun on non-Stanford models retrained on the augmented PTB + Stanford "additional examples." Errors are marked with asterisks, as before

Example	(4a)	(4b)	(4c)	(4d)	(4e)	(4f)	(4g)	(4h)	Correct
Berkeley	VB	VBN	VB	*VB	VBN	VB	VB	*VB	6/8
BC-M2	VB	VBN	VB	*VB	VBN	VB	VB	*VB	6/8
CJ-I	VB	VBN	VB	*VB	*VB	VBN	VB	*VB	5/8
Stanford-unlex	VB	VBN	VB	VBN	VBN	VBN	VBP	VBN	8/8
Stanford-lex	VB	VBN	VB	VBN	VBN	VBN	VBP	VBN	8/8

to examine what is causing the improvement, as was true in the wh-question case study. To examine this, we tested whether the 96 extra examples alone would suffice to correct some or most of the *read* errors. We therefore retrained all the parsing models, aside from CJ-R, using just the PTB training data plus the 96 "additional" examples, omitting the GENIA examples. We then re-ran the *read* example sentences, with the results shown in Table 5. There is an improvement in every case. Both Stanford parsers still have perfect scores, suggesting that the entire improvement is due to the 96 extra examples, rather than further additions from GENIA. Further, both the Berkeley, BC-M2. and CJ-I parsers improve, and now get 6/8 correct (they all fail on the third and the last *read* examples). We conclude that the judicious addition of even a few critical examples can greatly improve parsing performance, just as in the case of QuestionBank, again pointing to the sparsity of the original PTB training dataset as well as the ease with which some its failings may be remedied, at least in this particular situation.

However, it is still true that none of the systems explored here explicitly records the linguistic fact that the auxiliary at the front of the sentence is tied to the main verb. They do so only indirectly. Even in English, the properties of tense are "spread out" over the entire Auxiliary system. In an example such as *The stock could have been being sold*, it is the sequence of auxiliary verbs that together carry the tense information. It is only a morphological accident of English that these elements must generally be string-adjacent. Whenever two are separated by an intervening phrase, as in the *read* examples, the agreement between them still holds. It remains to be seen how to properly represent such facts in the statistically-grounded systems we have explored here.

Here we note that parameter estimation issues are a symptom rather than the underlying cause of the deficiencies of the parsing model. Such a model is unable to capture the interaction between wh-movement and the auxiliary/main verb system, or posit a connection from the declarative form of the sentence to its interrogative form without actually having observed the handpicked examples that closely match the test data.

5 Case Study: Parsing Passives by Linguistic Regularization

We noted in Sect. 1 that statistically-trained parsers make attachment errors in passive sentences, in part because attachment decisions are difficult without sufficient data. We also pointed out that in certain cases, this could be repaired by reconstructing a sentence's underlying "logical form" (a form of "D-Structure" in the classical sense), thereby rendering arguments in canonical positions. In general, we will call these kinds of reconstructions into a canonical predicate-argument form *linguistic regularizations*.

We note that several researchers have previously attempted to improve statistical parsing performance via representational changes to the grammar, in the form of either tree-level transformations, or by incorporating other latent information present in the Penn Treebank [7, 19, 25, 32]. Most of these approaches follow the paradigm proposed in [25], whereby the parser is retrained on a transformed version of the training set and then after evaluation the resulting parses are de-transformed and evaluated against the known gold standard annotations.

The approach we will take here differs from this past research in at least two critical respects. First, previous work such as that in [30] has focused on using additional features in the PTB as a means to improve parsing accuracy, while still others, as in [15] Chap. 7, model wh-displacements by means of feature passing. Few approaches have explicitly modeled a separate level of underlying predicate-argument structure. Second, more specifically, the level of syntactic complexity involved in these transformations has been rather limited, and none of the researchers up to the present point have attempted to reassemble the underlying representation of passive constructions.

Following the methodology of [25], we propose to exploit the additional information provided by linguistic regularizations in the following way. First, as suggested above, we can use the annotated PTB training trees to "invert" various displacement operations, returning arguments to their canonical "underlying" positions. In the case of our example sentence, we would derive something like, *Tablespoons may soon replace measuring cups in the laundry room.* We then use the transformed sentences as revised training data for a statistical parser. If the regularization idea is sound, then we would expect improved performance.

5.1 Passive Transformations: A Pilot Study

We will now show that employing "logical form" structural cues for linguistic regularization can improve parsing performance within the existing Penn Treebank formalism. We selected the passive because it has not, to our knowledge, been tackled in previous work. The experimental setup is as follows. As mentioned, we approach the problem within the framework proposed by Johnson [25]. We identify a set of transformations we would like to model in the corpus, transform the input

Table 6 Parsing results for models trained on the original (BASE) and transformed (TRANS) Penn Treebank (PTB) data. *untrans* corresponds to the untransformed or original corpus, while *trans* to the transformed version. *full* is the entire corpus; *psv*, the subset of passive sentences; *yactive*, the subset of active sentences. SBASE and STRANS experiments are oracle experiments – where the test set ("special") sentences are selectively transformed or kept intact to maximize the evalb recall. The POS column corresponds to the part of speech tagging accuracy. The size column identifies the number of sentences in the test corpus

Experiment id	Training set	Test set	Recall	Precision	POS	Size
BASE-1	wsj-02-21 untrans	wsj-23-full-untrans	88.17	88.36	96.87	2,416
BASE-2	wsj-02-21 untrans	wsj-23-full-trans	87.89	88.08	96.73	2,416
BASE-3	wsj-02-21 untrans	wsj-23-psv-untrans	87.75	87.96	97.40	364
BASE-4	wsj-02-21 untrans	wsj-23-psv-trans	86.28	86.43	96.65	364
BASE-5	wsj-02-21 untrans	wsj-23-active	88.27	88.45	96.75	2,052
TRANS-1	wsj-02-21 trans	wsj-23-full-untrans	88.26	88.48	96.86	2,416
TRANS-2	wsj-02-21 trans	wsj-23-full-trans	88.29	88.47	96.82	2,416
TRANS-3	wsj-02-21 trans	wsj-23-psv-untrans	87.39	87.65	97.27	364
TRANS-4	wsj-02-21 trans	wsj-23-psv-trans	87.51	87.62	97.02	364
TRANS-5	wsj-02-21 trans	wsj-23-active	88.46	88.66	96.77	2,052
SBASE	wsj-02-21 untrans	wsj-23-psv-special	88.12	88.22	97.02	364
STRANS	wsj-02-21 trans	wsj-23-psv-special	89.30	89.38	97.25	364

data by performing a set of deterministic 'tree' surgeries on the input parse trees, and then, after re-training, evaluate the resulting parser on a transformed test set.

The first step in this process is to perform tree regular expression (`tregex`) queries on the corpus to identify the passive constructions in the training data sections of the PTB. Second, we must map passive syntactic structures back into their active form counterparts. This mapping is achieved through a sequence of tree-transforms, applied recursively in a bottom-up, right to left fashion using the `Tregex` and `Tsurgeon` toolkit [31]. Note that in some cases, there will be no "by" phrase, that is, no explicit semantic Subject. In these cases, we insert a dummy subject with the part of speech label `TT`, corresponding roughly to *it*.

In all, there are 6,015 passive sentences in the training corpus out of a total of 39,832 sentences, or 15 % of the training data. In the test set, section 23 of the PTB corpus, 364 out of 2,416 sentences or 15.1 % of the test data can be identified as passives, comparable to the figures observed in the training set. The passive construction would therefore seem to provide a good test-bed for a pilot analysis. A ten percent sample of the identified training set items and all of the test set items were manually checked by a human expert who validated them as true passive constructions.

The third step of the procedure is to re-train and test a statistical parser on the transformed test and training data. We conducted our experiments using BC-M2 [3], following standard procedures. Additionally, we conducted our experiments on different combinations of transformed and untransformed training and test data, as well as allowing for configurations whereby the test corpora were evaluated on the active and the passive subsets separately. The pilot test results are displayed in Table 6.

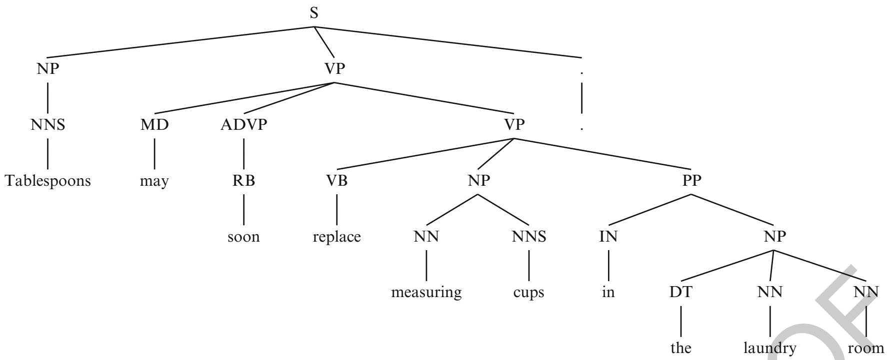


Fig. 8 The Bikel/Collins parser correctly analyzes the "tablespoon" sentence after regularization

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

On the full test set (2,416 trees), the retrained model (TRANS-2) beat the baseline (BASE-1) by 0.12 % absolute recall (88.29 % vs. 88.17 %) and 0.11 % absolute precision. On the active sentence subset that constitutes about 85 % of the test corpus, the model outperforms the baseline by 0.19 percent in recall – a statistically significant difference at the 0.05 level (p-value = 0.029) as computed by a stratified shuffling test with 10,000 iterations. While this may seem like a small performance gain, in the context of a trained parsing system that is known to be operating at close to a theoretical ceiling, this is in fact a real performance increase.

More concretely, to give an idea of an error that is corrected by regularization, in Fig. 8 we display the parser's output of the transformed example sentence, *Tablespoons may soon replace*... The parser outputs a tree that is 100 % correct.

To give a broader picture of where the performance improvement comes from, as another example, Fig. 9 displays an example from section 23 of the PTB, sentence #722, *According to analysts , profits were also helped by successful cost-cutting measures at Newsweek .*, that is parsed incorrectly in its unregularized form, with a misplaced PP high attachment for *at Newsweek*. This yields a labeled precision score of 91.67 % and a labeled recall score of 84.6 %. As the bottom half of Fig. 9 shows, after regularization this sentence is now parsed with perfect recall and precision, with a correct PP attachment under the NP.

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

However, the simple regularization carried out in the pilot study can sometimes also lead to worse performance: 95 out of 364 test sentence passives were

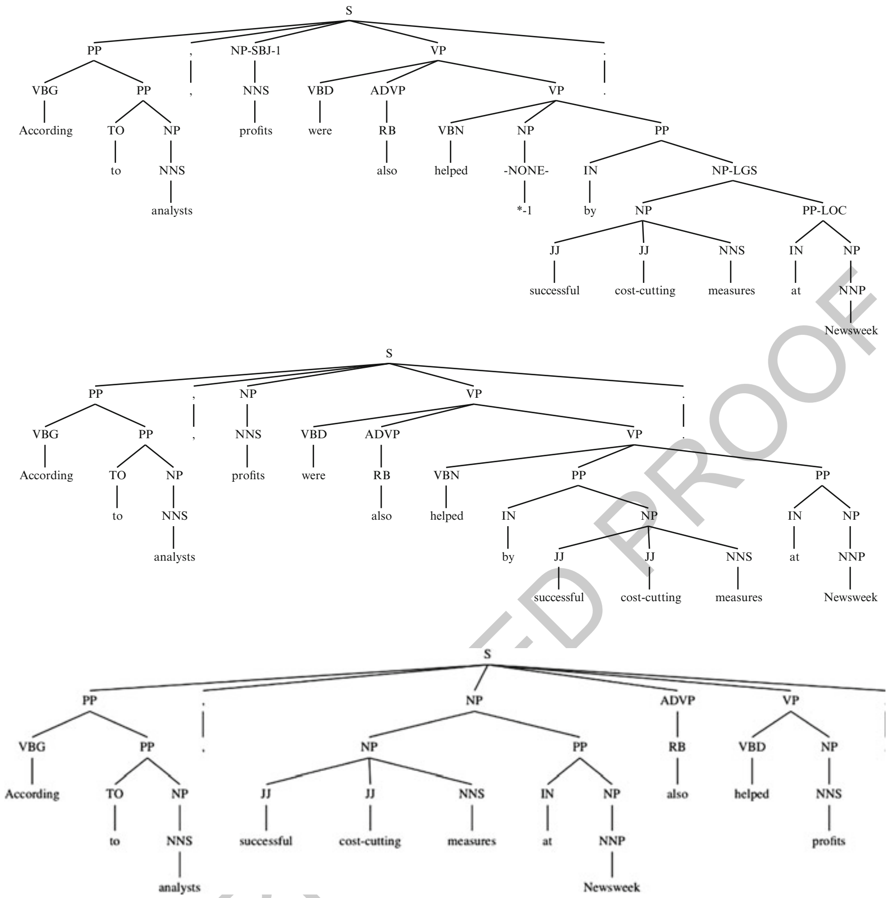


Fig. 9 The BC-M2 parser mis-analyzes of sentence #722 in section 23 of the PTB. The *top* third of the figure shows the gold standard parse. The *middle* third of the figure displays the corresponding (incorrect) BC-M2 parse. The *bottom* third shows the result of parsing the same sentence correctly after the regularization procedure described in the main text

parsed *worse* than before. It is these cases that reduce the performance gain of regularization in our pilot study. Figures 10 and 11 illustrate one example of this effect. Sentence #2,274 in test section 23, the passive sentence, *Tandem 's new high-end computer is called Cyclone*, is parsed with perfect precision and recall before regularization, though with an arguably incorrect gold-standard bracketing: both an empty Subject NP followed by a predicate NP *Cyclone* are dominated by an S. As Fig. 11 shows, after regularization, the re-trained parser mis-analyzes this structure with both the restored Subject NP *Tandem 's* and the predicate NP *Cyclone* combined as a single NP (precision = 71.43 %, recall = 83.33 %). It seems likely that examples such as these might be successfully analyzed if the gold-standard was assigned a linguistically more accurate "small clause" type structure.

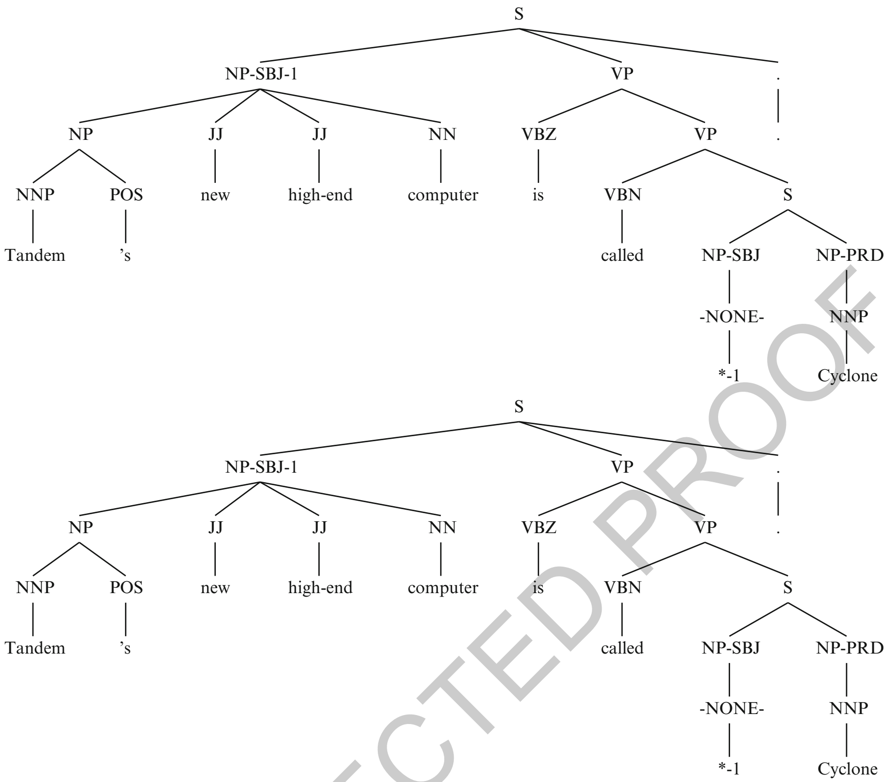


Fig. 10 The Bikel/Collins parser analysis of sentence #2,274 of section 23 of the PTB. The gold standard annotation is at the *top*, the parser output on the *bottom*

Other regularization failures occur where there is no following PP phrase in the original sentence to be mis-parsed, and where the regularization leads to a complex structure with the potential for misanalysis. For instance, the section 23 passive sentence #269, *The land to be purchased by the joint venture has n't yet received zoning and other approvals required for development , and part of Kaufman & Broad 's job will be to obtain such approvals .* requires the NP *the joint venture* to be restored as the Subject of *receive*. However, the re-trained parser incorrectly analyzes the regularized sentence. In part this may be the result of not completely reconstructing the underlying form; in this instance, where there is a relative clause *the land purchased by the joint venture*, the object of *receive*, *the land*, is not explicitly restored to its underlying position after the verb. Such complexity has tendency to lead to mis-analysis, and a more complete reconstruction of such relative clauses might repair such instances.

Note that even though on the passive subset (364 trees) the baseline outperforms the transformed model by 0.24 % recall, the result is not statistically significant (p-value = 0.295). Taken together, the results indicate that retraining significantly

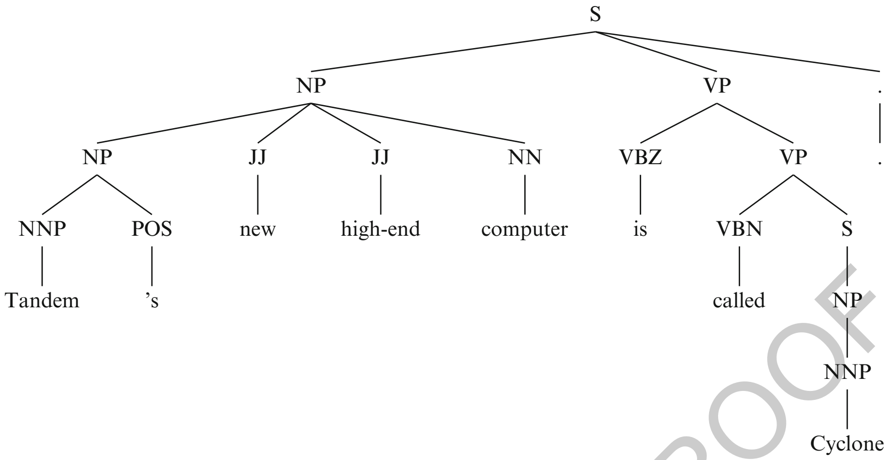


Fig. 11 The parse of regularized sentence #2,274 mis-analyzes the NP – NP structure under a single NP, precision = 71.43 %, recall = 83.33 %

improves the performance of the parser on active sentence constructions, while not incurring a statistically significant loss on passives. In fact, the retrained model is much more robust with respect to untransformed passives, only exhibiting a 0.12 % loss in precision, whereas the baseline suffers almost a 1.5 % degradation (TRANS-3 vs. TRANS-4).

We tested further potential for improvement by selectively unwinding certain passives into their underlying logical form, while leaving others in their original surface form. This is an oracle experiment, whereby we evaluate the parser only on the surface forms that achieve better performance under the retrained parsing model. That is, we assume the presence of an “ominiscient” selection procedure that allows us to decide whether the instance to be parsed for testing first needs to be transformed or whether it is more desirable to leave it in its original form. In carrying out the experiment we evaluated both forms for each test sentence and picked the one that achieved maximum evalb recall. Note that in practice, we would not have access to such a procedure. However, it is instructive to carry out this experiment, as it allows us to gauge the best possible (upper bound) performance for using an “unwound” logical form. This result indicates that we can obtain an upper bound of 89.30 % recall, as much as a full percentage point improvement over the baseline by applying the transformations on a selective basis. Further analysis of the results shows that this effect is achieved due to cases where displaced modifiers in the passive construction impact negatively on the parser’s attachment decisions.

Based on the evidence from the oracle experiment, we hypothesize that a simple binary classifier that could choose the training model from the features of the input test sentence should be able to recover much of the hypothetical gain due to the oracle.

Although seemingly small, the improvements obtained in the regularization experiments are statistically significant, and with more engineering effort in modeling nested passives and long-distance displacements we expect a greater gain.

We note that the important takeaway message from this pilot experiment is not that this is exclusively a parameter estimation problem. On the contrary, we point to the impracticality of adding a passive or active instance for every surface form observed in the training corpus without the extra linguistic knowledge explicitly encoded through structural transformations that map passive forms to their active counterparts. By incorporating linguistic knowledge we were able to improve a broken model indirectly by alleviating the parameter estimation problem.

By no means should this fix be viewed as a permanent solution. Our ability to make an impact suggests that the underlying representation is deficient and that much more radical changes need to be made to the model. One approach, by no means the only one, is by explicitly representing movement as a primitive operation. Alternatively, one could adopt a scheme like that of Combinatorial Categorial Grammar.

6 Parsing "Unnatural" Languages?

We turn in our final section to the Musso et al. experiment [36], in an attempt to probe to what extent statistically-based parsers can acquire "unnatural" language constructions. Recall from Sect. 1 that the second experiment in [36] was designed to see whether normal adults could easily learn a "mirror reversed" question formation rule, as well as whether this learning (as tested by subsequent parsing probes) activated the same brain regions, as visualized by fMRI. A typical example of such an natural/mirror-reversed pair, as cited earlier, is this: *il bambini amano il gelato*/*gelato il amano bambini il*. Their basic finding was that normal adults had extreme difficulty with such examples, solving them, if at all, as if they were non-linguistic puzzles, and drawing on different brain regions than those usually seen associated with language (specifically, outside Broca's area). Similar poor learning of "unnatural" language patterns has also been found in autistic language savants [49].

Our last experimental manipulation investigated whether we could replicate the second study described in [36] within the context of statistically-trained parsing. That is, we modified the PTB training data so that all question forms would be presented in their reverse or "mirror image" order, rather than in normal English word order. The parsers would then be trained on this manipulated data, and subsequently tested whether they had acquired the "mirror reverse question" construction by assessing them on a similarly question-reversed PTB section 23 data set.[9] In our emulation experiment, in addition to the standard PTB training

[9] We put to one side the question of carrying out fMRI experiments on computers.

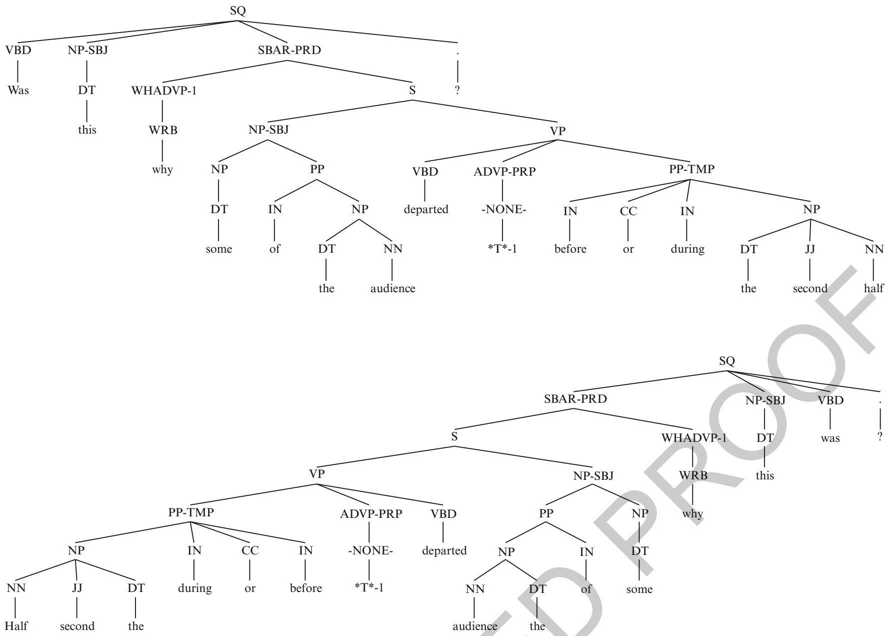


Fig. 12 Conventional and mirror-image treebank questions from the PTB, for training sentence#76, Was this why some of the audience departed before or during the second half?

sections, we also carried out a supplementary training/test regime again using the QuestionBank constructed by Judge et al. [26]. We did this because there are only 24 questions total in the entire standard test section 23 of the PTB, so that mirror-reverse questions are not properly exercised by the normal test dataset.

A typical example of such a "mirror image" training tree drawn from QuestionBank is displayed in Fig. 12 below, the mirror image corresponding to the question, *Was this why some of the audience departed before or during the second half?* Note that the input words are in reverse order (and the parse tree is the mirror reflection of the given parse tree in the treebank).

We should emphasize that there is a considerable challenge in carrying out this exercise properly in order to reflect (as it were) adult linguistic behavior and inference. It is, in general, not possible to exactly replicate the experimental conditions in [36]. The key problem is that we cannot be certain as to the internal system by which people processed the reversed sentences in [36]. As a first approximation, however, it may be fair to say that they could bring to bear the usual cognitive apparatus of "chunking" words into phrases (though the exact manner and details as to how much structural information is readily available remains a matter of some controversy; see [45], among much other recent work on this topic). However, it is reasonable to surmise that they did not have access to pre-formed parse structures, as is the case with the artificially constructed corpuses and the statistically-trained systems. In particular, in our emulation we gave the parsers the mirror-images of question sentences (including those embedded in quotational contexts), and one might reasonably object that this is far more information than that provided to the human subjects. This is a fair point. However, here we shall simply observe that [36] deliberately used Japanese (and German) native speakers for their experiments, just for this reason, since these languages are head-final, with left-branching structure similar to that displayed on the bottom half of Fig. 12, though of course not so uniformly reversed and not reversed solely with respect to questions. This was intended to compensate for any basic unfamiliarity with branching structures of the kind displayed in the figure, the implication being that these speakers would have had experience grouping lexical items in such a fashion. Further, this is evidence that intonational breaks to highlight structure and related cues are essential in some way for language inference in any case; see [35]. However, there is no denying that the exact experimental condition we used, providing both the reversed string and its corresponding mirror-image parse tree, has, to the best of our knowledge, never been replicated in any human subject experiment. This is true of many important questions regarding human language acquisition. For example, until it was first probed in [17], whether or not children actually formed Subject-Auxiliary verb questions using structural rules had not been experimentally addressed. Similarly, the question posed here is an empirical one that can only be resolved by future research.

6.1 The Experimental Emulation

To emulate the experiment in [36], we prepared two sets of training and test data, all with reversed questions, via manipulation of the PTB, along with the additional QuestionBank corpus. To start then, we had two training and two test datasets: (1) the standard training sections 02–21 of the PTB; (2) test section 23 of the PTB; (3) the normal training sections of the PTB concatenated with an 80 % sample of QuestionBank, 3,200 questions; (4) a held-out 20 % test sample of QuestionBank, 800 questions. (See Sect. 4 for a detailed description of QuestionBank.)

To obtain the appropriate mirror-image "reversed" question datasets we replaced all questions (both root level questions and questions in sentence contexts, usually quotational) in the original corpuses with their mirror-image counterparts. Figure 12 displays an example of a PTB training sentence #76 in its normal and mirror-reversed formats. The original sentence is, *Was this why some of the audience departed before or during the second half?*, while the reversed structure corresponds to, *Half second the during or before departed audience the of some why this was?* An example of a wh-question in a quotational context is sentence #610 of the training set, *"So what if you miss 50 tanks somewhere?" asks Rep. Norman Dicks, D., Wash., a member of the House group that visited the tanks in Vienna.* We carefully analyzed the original data to ensure that these were properly reversed. In this case, only the material within double quotes would be reversed.

For convenience, we will refer to all these training and test data sets along with their mirror-image question reversed counterparts as follows. There are four training sets in all, the two non-question reversed training sets and the two question reversed training sets. Similarly, there are four corresponding test sets. So altogether there are a total of 16 possible training-test dataset combinations. We will denote each of these training/test combinations with a unique label consisting of the training dataset name, a slash, and then the test dataset name. For example, WSJ/WSJT denotes the conventional WSJ training/WSJ section 23 test combination, while WSJR-QBR/QBRT denotes the WSJ training section with mirror-image questions augmented by the mirror-image questions as the training set, and the held-out mirror-image QuestionBank sentences as the test set. Note that the QuestionBank and the WSJ corpora are disjoint. The four training and four test sets are as follows.

1. **WSJ:** The conventional training sections 02–21 of the PTB;
2. **WSJR:** The question mirror-reversed training sections 02–21 of the PTB
3. **WSJ-QB:** The question-augmented corpus, sections 02–21 + the 80 % sample from QuestionBank;
4. **WSJR-QBR:** The question-reversed WSJ training section + mirror-reversed QuestionBank 80 % sample;
5. **WSJT:** The conventional test section 23 of the PTB;
6. **WSJT-R:** The question-reversed conventional test section 23 of the PTB;
7. **QBT:** The 20 % held-out test sample from QuestionBank;
8. **QBRT:** The question-reversed sentence test sample of QuestionBank.

6.2 Training, Testing and Results

We selected the BC-M2 and Stanford-unlex parsers as representative "lexicalized" and "unlexicalized" parsers for the experiment. Along with 16 training-test combinations, this yields 32 possible experimental runs. Note that four of these runs, the WSJ/QBT and WSJ-QB/QBT analyses for each parser, have already been carried out as part of the wh-QuestionBank testing in Sect. 3, but we include them below for completeness.

The results are summarized as F-scores in Tables 7 and 8. (We have split the results across two tables in order to highlight the most important contrasts in the first table.) The first table's results are also displayed in a more readable form as the histogram in Fig. 13, which presents F-scores on the Y-axis, and the most important training-testing contrasts on the X-axis; the BC-M2 results are in dark grey, and Stanford-unlex in light gray. Note that because there are so few questions in test section 23 of the PTB, just 20 out of 2,416 sentences, excluding a few non-question fragments that are marked as questions, that performance on the WSJ-T corpus does not serve as a reliable indicator of whether question sentences have been learned or not, though it may be of some value to see whether learning mirror-questions

Table 7 F-score results for the first eight training/testing results for the "mirror reversed" experimental manipulation. Lines (4)–(7) show that both lexicalized and unlexicalized parsers learn "mirror reversed" questions quite well

Train-test combination	BC-M2	Stanford-unlex
(1) WSJ/WSJT	85.63	85.54
(2) WSJ/WSJT-R	85.78	85.71
(3) WSJ/QBT	75.76	67.75
(4) WSJ/QBRT	13.15	19.12
(5) WSJR/QBRT	58.04	61.20
(6) WSJR-QBR/QBRT	65.94	71.47
(7) WSJR-QBR/QBT	55.67	60.58
(8) WSJ-QB/QBT	86.18	81.32

Table 8 The remaining 16 results for the WSJ "unnatural" learning experiments. Note that training by reversing just the questions in the WSJ, using WSJR, also boosts reversed-question parsing performance, but not as much as using the full training QBR training set. In general, testing on WSJR does not indicate any great difference, because there are so few questions in WSJT to test

Train-test combination	BC-M2	Stanford-unlex
(1) WSJ-QB/WSJT	85.79	81.32
(2) WSJ-QB/WSJT-R	88.01	85.46[a]
(3) WSJ-QB/QBRT	18.2	20.88
(4) WSJR/WSJT	85.63	85.54
(5) WSJR/WSJT-R	85.87	83.75
(6) WSJR/QBT	44.65	48.75
(7) WSJR-QBR/WSJT	85.59	85.19
(8) WSJR-QBR/WSJT-R	86.45	84.45

[a] We note that here both parsers do somewhat better on the mirror-image WSJT data than on the standard WSJT data when trained on QB, where one might expect the opposite result, but this difference may due to the sparse nature of the standard test section

interferes in some way with the parsing of normal based sentences. Therefore, we will in general put to one side comparisons based on just this test data set, e.g., contrasts like WSJ/WSJT vs. WSJ-QB/WSJT. We also leave for future research the measurement of statistical significance of the scores by means such as stratified shuffling, as in [3], or the assessment of oracle-type scores.

The key finding to take away from these results is that there is strong evidence that both parsers were able to learn the mirror-reversal question constructions quite well, though the lexicalized BC-M2 parser was less successful. To see this result most clearly one need only focus on the histogram bar marked with an arrow

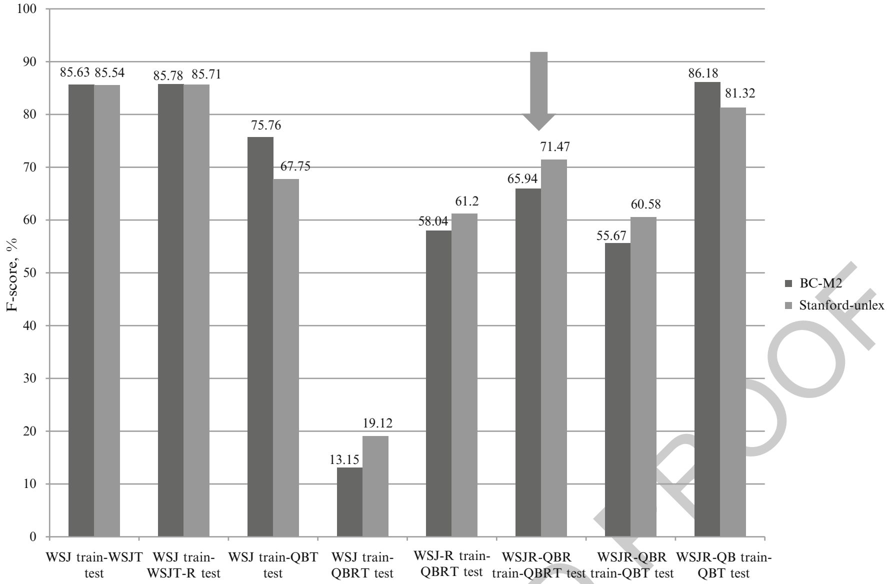


Fig. 13 F-score comparisons for BC-M2 and Stanford-unlex parsers show that the parsers do not perform well on mirror-image questions (the fourth, *middle* histogram pair from the *left*), but performance increased dramatically given QB mirror image question training, by 50 % points or more, as shown by the next two histogram pairs to the *right*. The right-most histogram repeats the finding from Sect. 3 showing that normal question parsing is also improved by the addition of normal QuestionBank training data

in Fig. 12, and note its performance gain compared to the preceding two bars, which summarize the before/after training effect. For example, when trained on only normal data, the Stanford unlexicalized parser scored only 19.12 % on the QuestionBank mirror-reversed test set, combination WSJ/QBRT, line 5 in Table 7 and the fourth histogram from the left in the figure. This number, then, may be taken as the "baseline" for a parser that has not learned anything about mirror-image questions. We may contrast this performance with training on just the WSJ reversed questions (which constitute only a small fraction, just few hundred examples out of nearly 40,000 sentences), line WSJR/QBR in the table. The initial 19.12 % figure goes up 50 % points, to 61.20 %, and additional QB mirror training examples boost this even further, another 10 % points, to 71.47 %, line 7, WSJR-QBR/QBR. Note that this is even better than the parser's performance on wh-questions after training on ordinary wh-questions. These are huge differences.

The performance gains for BC-M2 are nearly as good, though the actual numbers are less because the built-in English head-finding rules, which bias the formation of right-branching structures, cut against the grain of the mirror-reversed questions. Nevertheless, BC-M2 still performs remarkably well, as attested by examples like

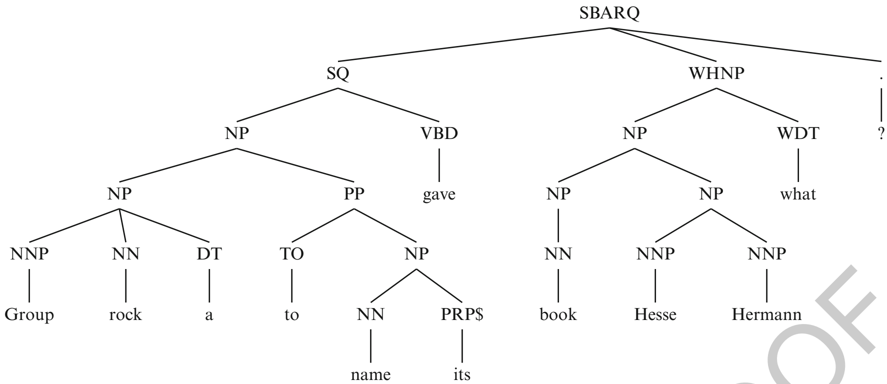


Fig. 14 BC-M2 correct parse of a "mirror" sentence from QuestionBank

the one shown in Fig. 14, the reversal of the QuestionBank sentence *What Herman Hesse book gave its name to a rock group*. Errors arise because the head rules attempt to locate heads at the left edge of phrases, except in Noun Phrases, but this of course is exactly opposite to what is required for mirror-reversed questions. A more careful experiment would re-do the BC-M2 head rules to locate heads at the right periphery, but one could then argue that we are in some sense aiding the parser in its discovery of the proper form for mirror-reversed questions. In a sense, it is startling that the BC-M2 parser works so well in spite of this handicap. Without any exposure to mirror-reversed questions, BC-M2 starts from a baseline of 13.15 %. This score rises to 58.04 %, line 6 in Table 7, a jump comparable to that of Stanford-unlex of more that 50 % in performance, after training on WSJ-TR examples. As with Stanford-unlex, training on reversed QuestionBank increases performance even further, to 65.94 % (line 7 in the table).

Row (7) and the next-to-last histogram bars in Fig. 13 the also indicate that the system has learned that questions are mirror-reversed: parsing performance drops by over 10 % when the systems are trained on WSJR-QBR, and then tested on normal questions, QBT. In short, there is every indication that mirror-image questions are learned with some facility.

It seems apparent that the BC-M2 parser could be further improved if the English-biased head-finding rules were re-written (though at the cost of "building-in" this linguistic knowledge). Figure 15 displays an example of a reversed sentence from QuestionBank, *What melts in your mouth not in your hands*, where the reversal, *Hands your in not mouth your in melts what* is given a (slightly) incorrect parse where a PP is mis-labeled as an NP. We will leave this more detailed analysis for future work.

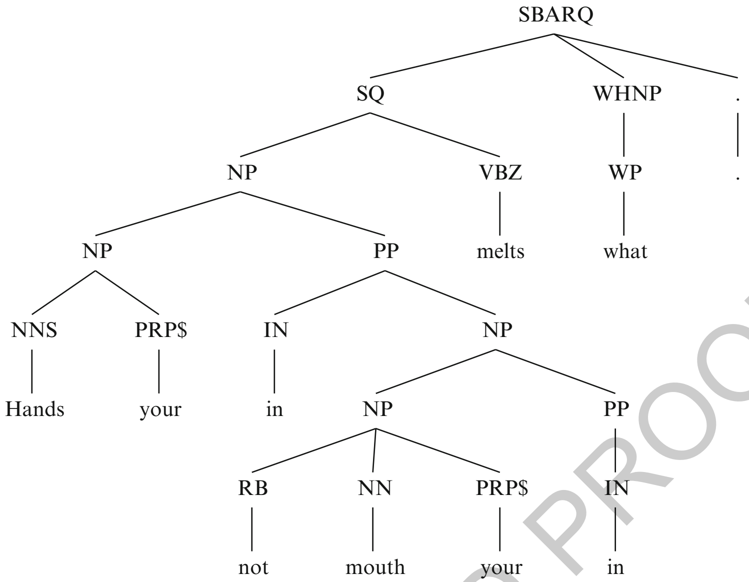


Fig. 15 BC-M2 parse of a "mirror" reversed question from QuestionBank with an erroneous labeling of a PP as an NP

7 Discussion and Conclusions

Let us now revisit the basic question outlined earlier and take stock of the results: Have state-of-the-art statistical parsers attained "knowledge of language"?

Current state-of-the-art systems, such as the several parser reviewed in this paper, score close to the 90 %-level (on withheld PTB data) when evaluated on phrase structure bracketing fidelity [16]. Of course, bracketing is not the only possible evaluation metric, as is now widely understood. In many cases, dependency relations may be of more importance; see [13] among many others for a discussion of this matter.

To the extent that such bracketing reflects linguistic knowledge, then such parsers do, of course successfully acquire that knowledge. Moreover, as noted by Petrov et al. [42] among others, modern statistical parsers can acquire tacit information about the details of verb subcategories, along with derivational structure. However, merely being able to bracket sentences "accurately" evidently does not constitute full "knowledge of language." Rather, knowledge of language is multi-dimensional and cannot be conveniently summarized in terms of a single number, an F-measure. Similarly, grammaticality cannot be described in terms of a simple probability score. We could not predict the outcome of the *read* experiment in advance simply by looking at aggregate F-measures, nor any other proposed measures we are aware of. Such conclusions may seem obvious from the outset, but the goal in applying the kinds of stress tests described in this chapter is to discover exactly where these systems fail.

The *read* sentences are also good exemplars of such a diagnostic aid. In this case, they point to a general issue with "long distance" agreement in tense (and other features) that is not to the best of our knowledge explicitly encoded in any of the statistical models, but only indirectly, perhaps through the use of extended horizontal and vertical domains of Markovization (as in the Stanford parsers), or through the use of latent variables. Even so, as we saw in the examples of the Berkeley and CJ systems with *read*, the use of tacit, indirectly formed categories may not precisely capture the right information. Rather, the results here suggest that it may be useful to explicitly import such machinery, as is done, for example, in the statistically-grounded versions of Lexical-Functional Grammar (see, for example, [43]; unfortunately, this system is not public and was not available to us for testing).

A second unsurprising result is that many of the limitations of current systems are due to the obvious sparsity of the PTB corpus. This effect is quite clearly displayed in the relatively poor performance on wh-questions, as well as how much that performance may be boosted by simply adding new wh-questions, sometimes only a handful, as the Stanford parser example illustrates.

In this chapter we have been able to select only one or two examples out of a long list of grammatical generalizations that linguists have accumulated over the past 60 years. It remains to analyze the remainder. The challenge for future research is whether these or similar diagnostics can be exploited to advance the state-of-the-art in statistical parsing. Given such a list, and given current statistical parsing methods based on discriminative methods, it may even be possible to construct a list of both positive and negative exemplars, as with minimally different wh-question examples, and then apply the method of "contrastive estimation" developed by Smith and Eisner [50] which compares positive training examples against negative examples in the local neighborhood of the training data. Some means of "discouraging" the leap to implausible or impossible word order patterns could be a welcome side-effect of this minimal use of negative examples, eliminating the ability to infer unnatural mirror-image structure.

The pilot experiment in Sect. 5.1 demonstrates that statistically significant improvements in parsing can be achieved by regularizing passive argument structure. However, in some cases passive regularization also led to worse performance. A more careful, case-by-case analysis of these examples would seem warranted. It appears from a superficial examination of the examples where parsing performance degrades that in each instance the regularization method has partly failed, sometimes introducing additional complex structure. If so, then further improvement may be possible if one can more accurately reconstruct the underlying form, either for small clauses or for relative clauses.

It seems clear that one could apply the notion of regularization more broadly to other types of displacements, such as topicalization and dislocation structures. We predict that these will provide additional parsing improvements, possibly approaching the levels achievable only through parse re-ranking. More generally, we note that the use of paired surface and underlying structures may provide great power not only in improving parsing, but also for providing a means to learn new rules to span the space of grammatical forms that have never been seen in training data, a major roadblock in state-of-the-art statistical systems. This is because our regularization approach bears important parallels to one of the few complete, mathematically established learnability results for a complete grammatical theory, that by Wexler and Culicover [53]. The Wexler and Culicover approach is based on a similar idea: the learner is assumed to be able to reconstruct the underlying "D-structure" corresponding to surface sentences, and from this pairing, hypothesize a possible mapping between the two. It remains for future research to determine whether this can be done for other displaced phrases in the PTB more generally.

Finally, we also note that in more recent grammatical theories, argument structure is regularized to an even greater degree by means of a VP-vP "shell structure" of branching nodes, that place Subject and then the Direct Object and Indirect Object NPs in specific, fixed positions with respect to the verb, perhaps in all languages [21]. If this is true, we could readily expand our regularization approach to this notion, which might provide a statistically-based, machine learning system with additional standardized patterns that are more easily learnable from training data alone. A full-blown incorporation of this kind of grammatical structure again remains for future work, but gives some hint at the untapped power of linguistic theory ready to be applied to treebank parsing.

Acknowledgements We would like to thank Michael ~~Coen,~~ Ali Mohammed for assistance and valuable suggestions. More importantly, we would like to extend special thanks to those individuals who have graciously made their parsing systems publicly available for open experimentation, in particular Daniel Bikel and Michael Collins; John Judge for his extremely valuable QBank resource and his generosity in providing it to us; Mark Johnson and Eugene Charniak; the members of the Stanford NLP group, including Daniel Klein and Christopher Manning; the Berkeley NLP group, including Stan Petrov and Daniel Klein; and the Malt and C&C parser developers. Without their generosity, analyses like those carried out here would be impossible. Finally, we would like to acknowledge two anonymous reviewers whose suggestions greatly improved this work.

References

1. Abney, S. (1996). Statistical methods and linguistics. In J. Klavans, & P. Resnik (Eds.), *The balancing act: Combining symbolic and statistical approaches to language* (pp. 1–26). Cambridge/Massachusetts: MIT Press.
2. Berwick, R. C., & Weinberg, A. S. (1982). *The grammatical basis of linguistic performance*. Cambridge: MIT Press.
3. Bikel, D. (2004a). *On the Parameter Space of Generative Lexicalized Statistical Parsing Models*. Ph.D. Thesis, University of Pennsylvania, Department of Computer Science.
4. Bikel, D. M. (2004b). Intricacies of Collins' parsing model. *Computational Linguistics, 30*(4), 479–511.
5. Charniak, E. (2000). A maximum-entropy inspired parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computtaional Linguistics* (pp. 132–139), Seattle. Association for Computational Linguistics.

6. Charniak, E., & Johnson, M. (2005). Coarse to fine *n*-best parser and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (pp. 173–180), Ann Arbor. East Stroudsburg: Association for Computational Linguistics.
7. Chiang, D., & Bikel, D. M. (2002). Recovering latent information in treebanks. In *Proceedings of the 19th International Conference on Computational Linguistics* (pp. 183–189), Howard International, Tapei.
8. Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.
9. Chomsky, N. (1968). *Language and mind*. New York: Harcourt-Brace.
10. Chomsky, C. (1969). *The acquisition of syntax in children from 5 to 10*. Cambridge: MIT Press.
11. Clark, S., & Curran, J. (2007). Wide-coverage efficient statistical parsing with ccg and log-linear models. *Journal of the Association for Computational Linguistics, 33*, 493–452.
12. Clark, A., & Lappin, S. (2009). Another look at indirect negative evidence. In *Proceedings of the EACL 2009 Workshop on Cognitive Aspects of Computational Language Acquisition* (pp. 26–33), Athens. Association for Computational Linguistics.
13. Clegg, A. B. (2008). *Computational-linguistic approaches to biomedical text sining*. Ph.D. thesis, Birbeck College, University of London.
14. Collins, M. (1997). Three generative, lexicalized models for statistical parsing. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics* (pp. 16–23), Madrid. Association for Computational Linguistics.
15. Collins, M. (1999). *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.
16. Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics, 29*(4), 589–637.
17. Crain, S., & Nakayama, M. (1987). Structure dependence in grammar formation. *Language, 63*, 522–543.
18. Curran, J., Clark, S., & Bos, J. (2007). Linguistically motivated large-scale nlp with C&C and boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions* (pp. 33–36),
AQ5 ~~Prague. Stroudsburg:~~ Association for Computational Linguistics.
19. Eisner, J. (2001). *Smoothing a probabilistic Lexicon via syntactic transformations*. Ph.D. thesis, University of Pennsylvania.
20. Gleitman, L., Gleitman, H., & Shipley, E. (1972). The emergence of the child as grammarian. *Cognition, 1*(2–3), 137–164.
21. Hale, K., & Keyser, S. (1993). On argument structure and the lexical representation of syntactic relations. In K. Hale, & S. Keyser (Eds.), *The view from building 20* (pp. 53–110). Cambridge: MIT Press.
22. Hockenmaier, J. (2003a). *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Doctoral Dissertation, University of Edinburgh.
23. Hockenmaier, J. (2003b). Parsing with generative models of predicate-argument structure. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 359–366), ~~Sapporo. East Stroudsburg~~: Association for Computational Linguistics.
24. Jackendoff, R. (1999). *Why can't computers use English?* New York: Linguistic Society of America (LSA) Publications.
25. Johnson, M. (1998). Pcfg models of linguistic tree representations. *Computational Linguistics, 24*(4), 613–632.
26. Judge, J., Cahill, A., & van Genabith, J. (2006). Questionbank: Creating a corpus of parse-annotated questions. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics* (pp. 497–504), ~~Sydney. Stroudsburg~~: Association for Computational Linguistics.
27. Klein, D., & Manning, C. (2003a). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 423–430), Sapporo. East Stroudsburg: Association for Computational Linguistics.

28. Klein, D., & Manning, C. (2003b). Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems* (pp. 3–10), Cambridge.
29. Lappin, S., & Shieber, S. M. (2007). Machine learning theory and practice as a source of insight into universal grammar. *Journal of Linguistics, 43*(2), 393–427.
30. Levy, R. (2006). *Probabilistic models of word order and syntactic discontinuity*. Ph.D. thesis, Stanford University.
31. Levy, R., & Andrew, G. (2006). Tregex and tsurgeon: Tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, Genoa.
32. Levy, R., & Manning, C. D. (2004). Deep dependencies from context-free statistical parsers: Correcting the surface dependency approximation. In *Procedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 327–334). East Stroudsburg: Association for Computational Linguistics.
33. Marcus, G. (2003). *The algebraic mind*. Cambridge: MIT Press.
34. Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1994). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics, 19*(2), 313–330.
35. Morgan, J., Meier, R., & Newport, E. (2004). Facilitating the acquisition of syntax with cross-sentential cues to phrase structure. *Journal of Memory and Language, 28*(3), 360–374.
36. Musso, M., Moro, A., Glauche, V., Rijntjes, M., Reichenbach, J., Buchel, C., & Weiller, C. (2003). Broca's area and the language instinct. *Nature Neuroscience, 6*, 774–81.
37. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kubler, S., Marinov, S., & Marsi, E. (2007). Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering, 13*(2), 95–135.
38. Nivre, J., Rimell, L., MacDonald, R., & Rodriguez, C. G. (2010). Evaluation of dependency parsers on unbounded dependencies. In *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing. International Association for Computational Linguistics.
39. Parisse, C. (2012). Rethinking the syntactic burst in young children. In A. Alishahi, T. Poibeau, A. Korhonen, & A. Villavicencio (Eds.), *Cognitive aspects of computational language acquisition*. New York: Springer.
40. Petrov, S., & Klein, D. (2007). Learning and inference for hierarchically split PCFG's. In *AAAI 2007 Nectar Track*, Washington. AAAI.
41. Petrov, S., & Klein, D. (2008). Sparse multi-scale grammars for discrimininative latent variable parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 867–876), Honolulu. Association for Computational Linguistics.
42. Petrov, S., Barrett, L., Thibaux, R., & Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics* (pp. 433–440), ~~Sydney. Stroudsburg~~: Association for Computational Linguistics.
43. Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell, J. T. I., & Johnson, M. (2002). Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)* (pp. 271–278), Philadelphi~~e~~.
44. Rimmell, L., Clark, S., & Steedman, M. (2009). Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Meeting on Empirical Methods on Natural Language Processing* (pp. 813–821), ~~Singapore. Morristown~~: Association for Computational Linguistics.
45. Saffran, J., & Newport, E. (2007). Statistical learning in 8-month old infants. *Science, 274*(5294), 1926–1928.
46. Sekine, S., & Collins, M. (2008). The evalb program.
47. Shipley, E., Smith, C., & Gleitman, L. (1969). A study in the acquisition of language: Free responses to commands. *Language, 45*, 322–343.
48. Smith, N., & Johnson, M. (2007). Weighted and context-free grammars are equally expressive. *Computational Linguistics, 33*(4), 477–491.

49. Smith, N., Tsimpl, I. -M., & Ouhalla, J. (1993). Learning the impossible: The acquisition of possible and impossible languages by a polyglot savant. *Lingua, 91*, 279–347.
50. Smith, N. A., & Eisner, J. (2005). Guiding unsupervised grammar induction using contrastive estimation. In *International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Grammatical Inference Applications* (pp. 73–82), Edinburgh~~.~~
51. Tateisi, Y., Yakushiji, A., Ohta, T., & Tsujii, J. (2005). Syntax annotation for the genia corpus. In *Proceedings of the International Joint Conference on Natural Language Processing* (pp. 222–227), JJeju Island~~.~~
52. Turian, J., & Melamed, I. D. (2006). Advances in discriminative parsing. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics* (pp. 873–880), ~~Sydney. Stroudsburg:~~ Association for Computational Linguistics.
53. Wexler, K., & Culicover, P. (1983). *Formal principles of language acquisition*. Cambridge: MIT Press.

AUTHOR QUERIES

AQ1. The unnumbered quotation has been linked with numbered citation (Ex. 1) in most of the occurrences in this chapter. Please check if okay.
AQ2. The section wise numbering of citations have been changed to continuous numbering for Tables. Please check if okay.
AQ3. Please provide closing parenthesis in the sentence starting "For example, *read*...".
AQ4. Table 4 has been changed to Table 5 as per MS. Please check if appropriate.
AQ5. Please check if inserted publisher location for [18, 23, 26, 32, 42, 44, 52] is okay.