# ADVERSARIAL TESTING OF STATISTICAL PARSERS:

# THE CASE OF TEMPORARY AMBIGUITIES

Sandiway Fong

University of Arizona

sandiway@arizona.edu

## Abstract

Garden-path sentences, containing a temporary attachment ambiguity of a noun phrase as either the object or subject of a sentence, are often studied in psycholinguistics because they can reveal whether verb subcategorization, semantics, or pragmatics are employed as part of on-line human parsing. We propose that these experiments can jump the human/machine divide and be applied for adversarial testing of broad-coverage dependency-based parsers for which little is known outside of black-box testing. In particular, here we use experimentally verified garden-path sentences to examine whether such systems have acquired cognitively accurate knowledge of language. This reveals a surprising lack of cognitive knowledge of certain basic aspects of English, as well as a contrast between the performance of these systems and traditional probabilistic context-free parsers.

## INTRODUCTION

The introduction of *neural net* (NN) transition-based dependency parsing systems has revolutionized statistical parsing. Parsing models based on classic probabilistic context-free grammars (PCFG) have been largely abandoned in favor of statistical systems that train on and directly compute dependency parses. The availability of dependency-based treebanks for a broad variety of languages, and the state-of-the-art scores reported underscore the motivation for this shift. For example, Google's Syntaxnet (Andor *et al*., 2016), achieved an *unlabeled attachment score* (UAS) of 94.6 on the Wall Street Journal (WSJ) corpus, adapted from the Penn Treebank (PTB) (Marcus et al., 1993).[1] Moreover, Google's initial claims of "*the world's most accurate parser*" and "*we are approaching human performance*" (Petrov 2016) have drawn considerable attention both within the natural language processing community and from a wider audience:

> "*We want to encourage the research community* [...] *to move beyond parsing, towards the deeper semantic reasoning that is necessary… We're basically telling them*: *You don't have to worry about parsing. You can take that as a given*."
> (Pereira, F., quoted in "Google Has Open Sourced SyntaxNet"; *Wired Magazine*, 2016.)

The claim of "approaching human performance levels" merits further investigation. For example, do these NN-based parsers achieve their impressive results with or without the same kind of knowledge of language that we know humans employ? We note that there has been prior interest in this question, e.g., Kuncoro *et al*.'s (2017) finding that *recurrent* NN *grammars* (RNNGs) seem to acquire the human language property of endocentricity for some phrases, Wilcox *et al*. (2018) on filler-gap restrictions, and McCoy *et al.*'s (2018) on subject-verb agreement, among much other recent work.

Here we propose a different type of "cognitive fidelity" test: that psycholinguistics data can be used to fruitfully probe the knowledge acquired by these nearly opaque NN-based systems. We place this research

[1] Scores for dependency parses are not directly comparable to their phrase structure counterparts: the former computes a figure of merit denominated in terms of word-to-word relations, the latter in terms of phrasal bracketing.

The author is indebted to Robert Berwick, Noam Chomsky and Riny Huybregts for their comments and advice on earlier drafts of this paper. An earlier version of this paper was presented at the Generative Grammar at the Speed of 90 workshop, University of Arizona, December 17th 2018. This draft: Sept 2 2020.

squarely within the framework of *adversarial testing for machine learning* (ML), beyond conventional tests of *n*-fold cross validation over standard corpora.[2] The need for adversarial testing as simply a part of basic "stress test" engineering practice has recently been recognized across ML domains.

Algorithmic adversarial testing is now well-accepted in image recognition and other domains where security is at stake, e.g. Kurakin *et al*. (2016). For language, the issue is perhaps even more acute, as a result of language's open-ended, infinite productivity, a property recognized at least since Humboldt (1836). Algorithmic adversarial testing has also been explored for language, e.g. Cheng *et al*. (2018) in the case of seq2seq NN models. We argue here in the case of language that well-understood, adversarial data already exists, not only for the purpose of stress-testing, but also with the real advantage that such data has been designed to reveal human knowledge of language.

In this paper we consider a concrete example of such pre-existing adversarial data. We use Traxler's (2002) psycholinguistics experiment, designed to elicit whether humans make use of their knowledge regarding syntactic verb subcategorization, e.g., transitivity, along with semantic/pragmatic information about verb-object compatibility in deciding whether to attach a parsed noun phrase (NP) as a *direct object* (dobj) of a verb, or alternatively, as the *nominal subject* (nsubj) of a following verb. By manipulating the initial verb and the (temporarily) ambiguous NP, we know it is possible to spoof humans into making the wrong initial parsing attachment decisions; hence, the descriptive term *garden path sentence*. Of course, humans go on to recover and assign the proper parses; i.e. the interplay between local attachment and global information about sentential phrase structure will be correctly resolved by the human parser. But what can we expect Google's system to do? When supplied with the complete sentence, will the temporary ambiguity not affect the parser at all? Or could the parser be permanently misled by local attachment preferences?

Here we report the results of analyzing Traxler's data on 3 dependency-based parsers, comparing the results to 2 reliable *probabilistic context-free grammar* (PCFG) parsers. Traxler's data set consists of a total of 78 garden path sentences with simple vocabulary (Traxler 2002).[3]

We exploit the fact that Traxler's sentence design has already controlled for various factors, so that it has already been designed as a set of adversarial examples (for human parsers). We test these adversarial sentences on the computer parsers. We conclude that the Traxler data do in fact create problems for the NN-based models that human parsers do not exhibit, suggesting that these deep learning models have missed learning some surprisingly simple facts about human language, despite their large training sets and high conventional accuracy scores. Moreover, somewhat surprisingly, one of the older, baseline PCFG parsers achieved a perfect score on the Traxler-designed sentence tasks. We performed an additional perturbation experiment to confirm that perfect performance was purely artifactual.

## BACKGROUND

First, some preliminaries: a dependency parse is a connected, acyclic set of directed arcs encoding syntactic dependency relations between words in a sentence, e.g. as in Figure 1(a). ML requires pre-analyzed sentences, and (native) dependency treebanks now exist for a variety of languages. Software also exists for transforming constituent-based parses into dependency parses, allowing NN-based systems to take advantage of the Penn

[2] Conventional tests draw from the same materials as training data, and thus make the i.i.d. assumption that test data "looks the same" as the training data. In effect, the test data check for good interpolation, but not for true generalization or extrapolation outside the training domain.

[3] A reader unfamiliar with psycholinguistic experimental design might question whether 78 examples are sufficient in this day and age of large corpora. With regard to experiment design, the number of sentences used (26 × 3 types = a total of 78) suffices to reveal statistically significant differences between human test subjects. Moreover, the constructed sentences, i.e. not chosen out of corpora, are carefully vetted for confounds. In fact, the original experiment was designed to test for child and adult language differences; hence also the simple vocabulary, which should pose no problems for the sophisticated word embeddings used in NN-based parsers.

Treebank (PTB), which precipitated much of the work in broad-coverage PCFG parsing in the 1990s.[4] For example, Figure 1actually re-encodes the PTB tree given in Figure 2 using Universal Dependencies (UD).

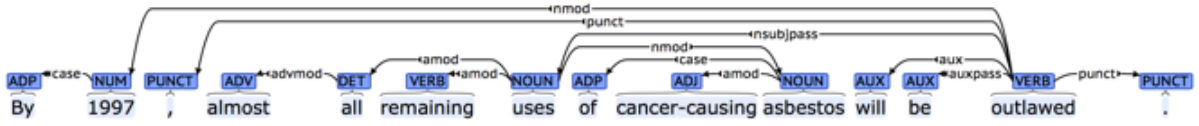


**FIGURE 1: AUTOMATICALLY CONSTRUCTED DEPENDENCY PARSE FOR THE PTB EXAMPLE GIVEN IN FIGURE 1(B)**

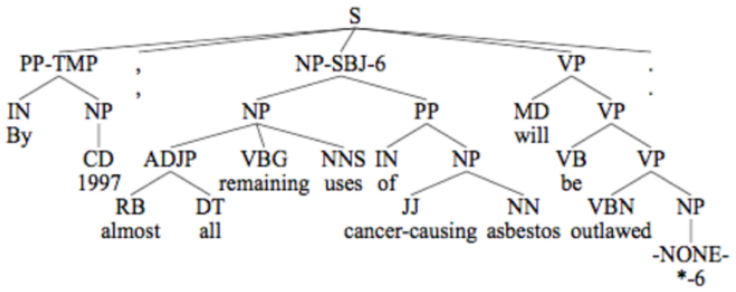


**FIGURE 2: PTB PARSE OF "BY 1997, ALMOST ALL REMAINING USES OF CANCER-CAUSING ASBESTOS WILL BE OUTLAWED."**

A transition-based dependency parser constructs a parse by processing input sentences a word at time from left to right. Similar in concept to shift-reduce (phrase structure) parsing, a transition-based dependency parser shifts words into a working memory buffer, usually encoded as a stack, and creates arcs between words in working memory, popping off dependent words from the stack as required; e.g., see (Nivre 2006).[5] An oracle, trained on labeled data, navigates the space of possible shift/arc operations depending on parser state; i.e. it decides when to shift and what dependency attachments to make.[6] In this paper, we will be focusing on dobj vs. nsubj attachments. The oracle will be a NN-based model, and words are first encoded using dense (low-dimensional) numeric vectors, called word embeddings; for example, as in (Chen & Manning 2014).[7]

The Universal Dependencies (UD) project is an attempt to develop a cross-linguistic set of relations for both annotation and parsing, accessible to non-linguists, see (Nivre 2015). In this paper, we limit our attention to a subset of the available UD relations including a few legacy relations (used by Google), summarized in Figure 3.

---

[4] For example, Stanford's UniversalDependenciesConverter may be used to compute a projective dependency parse from any PTB tree, with arc labels mapped to some UD set. The accuracy of the mapping is of vital importance. In principle, it should also be possible to produce non-projective dependency parses containing discontinuous elements, e.g. empty nodes labeled as *Interpret Constituent Here* (*ICH*) or *Right Node Raising* (*RNR*). Note that the reverse direction, i.e. dependency to constituency parse, is non-deterministic, as there are multiple possible constituent parses for any non-trivially sized dependency parse; see (Kong et al. 2014) for an example of a machine-learning approach to the problem.

[5] We have intentionally glossed over some differences in transition-based models; for example, *arc-standard*, i.e. attach a dependent only when the dependent itself has no dependents, vs. *arc-eager*, attach dependents early but keep them on the stack for further possible modification.

[6] The parser state that the oracle may condition its decision on will typically include a fixed (in length) prefix of input words, their part-of-speech (POS) tags, stack elements and associated arcs.

[7] Chen & Manning (2014) employed not only (pre-trained from large text corpora) word embeddings, but also part-of-speech (POS) and dependency relation embeddings.

| NP-related | Description | Clausal relations | Description |
|---|---|---|---|
| *nsubj* | nominal subject | *advcl* | adverbial clause modifier |
| *dobj* (v1), *obj* (v2) | direct object | *vmod* (SD) | non-finite verbal modifiers, superseded by *advcl* |

**FIGURE 3: SOME UNIVERSAL DEPENDENCY (UD) RELATIONS.[8]**

## THE TEST DATA

Traxler (2002) describes psycholinguistic experiments involving three kinds of temporary parsing ambiguities that arise when reading examples of the sort illustrated in examples (1-3):

(1) As the woman was cleaning *the stove* began to heat up.
(2) When Sue tripped *the table* fell over and the vase was broken.
(3) When the tiger appeared *the lion* roared very loudly.

Although all three examples are (ultimately) grammatical, measurable differences in reading times occur near the *italicized* NP above. Since human sentence processing is on-line, a temporary ambiguity is introduced just after the NP *the stove* in (1), initiated by the presence of the verb *began*.[9] Any initial attachment of *the stove* and *the table* as the dobj of *cleaning* in (1), and *tripped* in (2), respectively, must be reassessed when the rest of the sentence is presented, and the NP re-attached as the nsubj of the second verb, *viz*. *began* in (1) and *fell* in (2). However, the *italicized* NP is a plausible dobj of the initial verb in (1), *cleaning the stove*, but not in (2), *#tripped the table*. In (3), where the verb *appear* does not take a surface object, one may question whether there is an initial reflexive attachment of *the lion* as the dobj of *appear* at all.[10] The question explored by Traxler's experiments is whether information about subcategorization and thematic/semantic mismatches measurably affects the cost of human parsing.[11]

Focusing now on dependency parsing, the expected behavior can be schematized as in Figures 4 and 5.

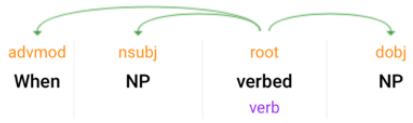


**FIGURE 4: INITIAL ATTACHMENT OF NP AS DOBJ.**

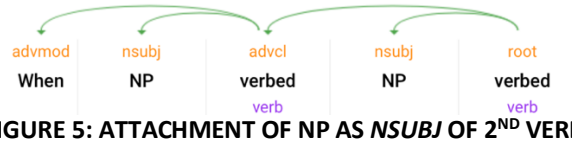


**FIGURE 5: ATTACHMENT OF NP AS *NSUBJ* OF 2ND VERB AND 1ST VERB AS *ADVCL* OF 2ND VERB.**

In Figure 4, the relevant NP is attached to the root verb as the dobj.[12] With the second verb present in Figure 5, there are two significant differences: (i) the NP becomes the nsubj of the second verb, the (new) root (or

[8] In Figure 2, SD refers to the legacy Stanford Dependencies pre-dating the UD project; v1 and v2 refer to UD version 1 and 2, respectively.
[9] An appropriate pause or punctuation, depending on the modality, before the highlighted NP will disambiguate the sentence.
[10] *Appear* is an example of an unaccusative verb, i.e., a verb with an underlying object that surfaces as a subject. In the technical literature, unaccusatives are contrasted with unergatives, e.g. unergative *sing* as in *John sings*, that contain no underlying object. Both unaccusatives and unergatives are examples of intransitives, but considered structurally distinct.
[11] More precisely, Traxler is interested in the difference between child and adult human processing for examples of this sort. We refer the interested reader to the cited paper. This is not a distinction that will concern us.
[12] For well-formedness, the NP must be integrated into the parse, i.e., cannot be left dangling. Therefore the oracle selects dobj as the most likely syntactic relation. Note also that, currently, Google emits the UD v1 relation dobj, which has been superseded in UD v2 by the more general *object* (obj), as not all languages exhibit the dobj–iobj pairing.

matrix verb), and (ii) the first verb is no longer root, but relegated instead to be an *adverbial clause modifier* (advcl) of the second verb.

Let consider some of the possible parsing errors. Figures 6 and 7 illustrates two possible mis-parse scenarios. Let us call the error in Figure 6 a dobj error, i.e. a failure to shift the dobj NP from the first verb to the second verb. Call Figure 7 an advcl error, i.e. a failure to shift root onto the second verb and mark the first verb as an advcl. The two errors are independent: there is a third logical possibility in which they occur simultaneously.

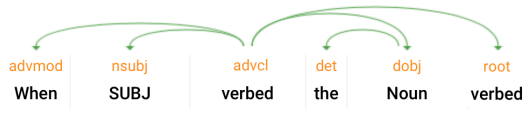


**FIGURE 6: DOBJ ERROR**

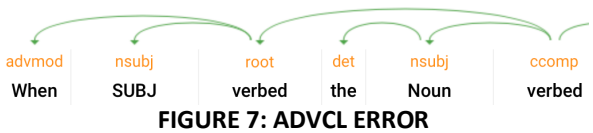


**FIGURE 7: ADVCL ERROR**

Correctly decoding the argument structure of sentences is at the heart of any syntactic parser, so the correct attachment of dobj and advcl relations is of one of vital importance, not just for the case of temporary ambiguities, but for dependency parsing in general.[13]

## RESULTS AND DISCUSSION

Traxler's experiments used 26 examples for each of three types of temporary ambiguity: *viz*. (i) a plausible dobj and a transitive (tr.) verb, (ii) a semantically implausible dobj and a tr. verb, and (iii) an intransitive (intr.) verb. These three types. illustrated previously as (1)-(3), are now annotated as (4-6) below.

(4) Experiment (tr. verb, plausible dobj):
As the woman was cleaning *the stove* began to heat up.
(5) Experiment 2 (tr. verb, implausible dobj):
When Sue tripped *the table* fell over and the vase was broken.
(6) Experiment 3 (intr. verb):
When the tiger appeared *the lion* roared very loudly.

The results obtained by parsing Traxler's 78 (26x3) constructed examples using Google's dependency parser are displayed in Figure 8.[14] The percentage of correct parses ranges from 62% in Experiment 1 up to 73% in Experiment 3. Error rate-wise, we obtain the ordering: intr. verb < implausible dobj < plausible dobj, which suggests partial accounting for both verb valency and dobj plausibility.

| Experiment | 1 | 2 | 3 |
|---|---|---|---|
| correct: | 62% | 65% | 73% |
| advcl | 27% | 15% | 19% |
| dobj+advcl | 11% | 12% | 0% |
| dobj | 0% | 8% | 8% |

**FIGURE 8: GOOGLE PARSER RESULTS**

[13] Other types of attachment errors are important too, e.g. for core relations like nsubj, and modification, e.g. preposition phrase (PP) attachment. However, we focused on the dobj/advcl paradigm, and did not penalize the tested systems for other kinds of syntactic errors.

[14] We report results for the commercial version as it is the best available version of the DRAGNN parser, short for Dynamic Recurrent Acyclic Graphical Neural Networks, described in (Kong et al. 2017). The commercial version has access to additional training data (not available to the research community), and also outperforms the earlier SyntaxNet parser. (Results were obtained in Oct. 2018 via cloud.google.com/natural-language/.)

Note the majority of the errors can be categorized as advcl errors, schematized in Figure 4(b), i.e., the parser (mistakenly) believes the first verb is the matrix verb and that the second verb actually heads a dependent clause, typically of type *complement clause* (ccomp).

An example of an advcl-only error is given in Figure 9. Although the parser correctly attaches *the song* as the subject of *play*, it incorrectly asserts that *sing* subcategorizes for a sentential complement; e.g. as if *sing* in (7a) was parallel to *think* in (7b).

(7) a. While the man *sang* the song was playing on the radio
b. The man *thought* [$_{CP}$ that the song was playing on the radio].


**FIGURE 9: ADVCL ERROR FROM EXPERIMENT 1.**

Figure 10 illustrates a Google parser dobj (only) error from Experiment 3, i.e., with an intransitive verb. Although the parser correctly indicates that *sob* heads a clause that modifies the main clause headed by *give*, it fails (as English is not a language that generally allows empty subjects) to assign an overt subject for the main clause, preferring instead to analyze *the man* as the direct object of *sob*.

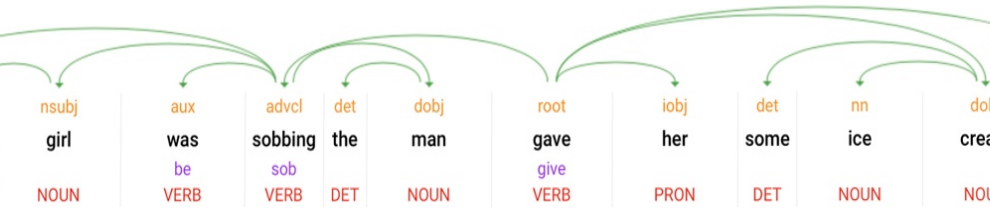


**FIGURE 10: DOBJ ERROR FROM EXPERIMENT 3.**

Finally, Figure 11 contains an example of a double error, produced by the Google parser on Experiment 2, paraphrased as (8).

(8) When the captain was sailing the truck crossed over the bridge.

In other words, not only did the parser incorrectly make *the truck* the direct object of *sail*, it also indicated that *sail* should head the sentence. Then *cross over the bridge* is relegated to be a modifier of *truck.*

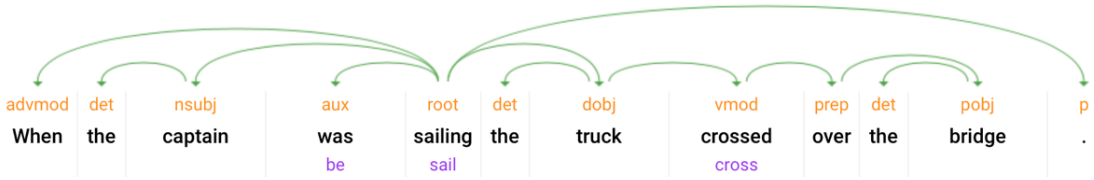


**FIGURE 11: COMBINATION ADVCL AND DOBJ ERROR FROM EXPERIMENT 2.**

Although it should be clear that the claim of "*approaching human performance*" is not substantiated, it is difficult to assess the 62–73% parsing results *in vacuo*. To provide some context, Figure 12 compares Google to AllenNLP and UDPipe, two other examples of NN transition-based dependency parsers.

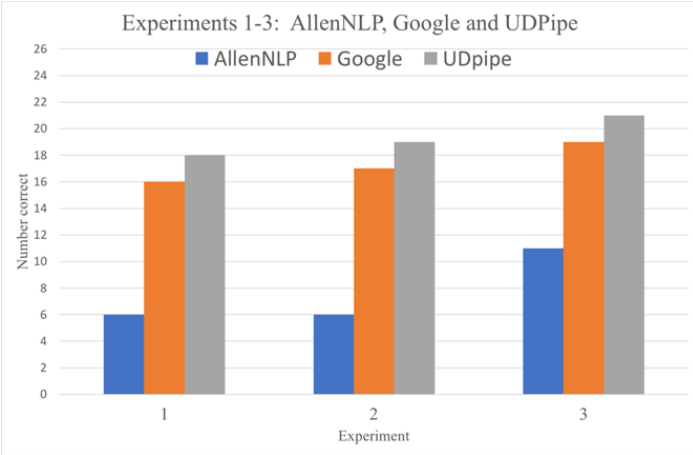


**FIGURE 12: A COMPARISON OF ALLENNLP, GOOGLE AND UDPIPE ON EXPERIMENTS 1–3.**

Although AllenNLP claims state-of-the-art PTB WSJ performance, achieving scores of 95.6 (UAS) and 94.4 (LAS, *labeled attachment score*) (Dozat & Manning 2017), it performed poorly on these experiments, scoring just 23–43%. We omit further discussion of AllenNLP. Figure 13 indicates that UDPipe performs similarly to Google, outpointing it 69–81% vs. 62–73%.[15] However, this is not statistically significant using Fisher's exact test. More noteworthy is that similar performance was achieved with varied training data.[16] Next, Figure 10 breaks out the error types for UDPipe, *cf.* Figure 8 for Google.

| Experiment | 1 | 2 | 3 |
|---|---|---|---|
| correct: | 69% | 73% | 81% |
| advcl | 8% | 12% | 0% |
| dobj+advcl | 15% | 4% | 4% |
| dobj | 8% | 11% | 15% |

**FIGURE 13: UDPIPE PARSING RESULTS.**

[15] The model english-ud-2.0-conll17-170315 was used.

[16] UDPipe uses UD v2 treebanks for training exclusively (Straka & Straková 2017). In particular, UDPipe does not make use of the PTB corpus mentioned in §2. We believe Google makes use of the PTB plus other corpora, some proprietary (as mentioned earlier).

There were no overlaps (with respect to errors) between the two systems in Experiment 3, but both systems made the same double error when the first verb was *sail* in Experiments 1 & 2, see (9a-b).

(9) a. When the captain was sailing *the ship* passed under the bridge. (Experiment 1)
b. When the captain was sailing *the truck* crossed over the bridge. (Experiment 2)

In either case, the preferred matrix verb, was *sail,* and the *ship*/*truck* was analyzed as its dobj. And (*passed under*/*crossed over*) *the bridge* was attached as a modifier of the dobj.

Finally, let us compare these deep learning parsers to the (now) research-orphaned PCFG-based systems. We selected the Stanford and Berkeley PCFG parsers (Petrov & Klein 2007).[17] In the case of the Stanford parser, we tested both the unlexicalized PCFG model and the lexicalized PCFG, henceforth LPCFG, model (Klein & Manning 2003). A "vanilla" PCFG is limited to the inventory of non-terminals and POS tags present in the training treebank. Simply put, a LPCFG expands a PCFG by incorporating head words directly into PCFG rules, thereby permitting finer-grained distinctions than POS tags alone (at the cost of requiring more training data).

Figure 14 summarizes the performance of all the parsers mentioned in this paper across the three Traxler experiments. We observed a large difference in performance between the two Stanford models. The unlexicalized parser compares poorly, similar in level to AllenNLP, but, rather surprisingly, the LPCFG parser is statistically indistinguishable from the modern Google and UDPipe parsers.

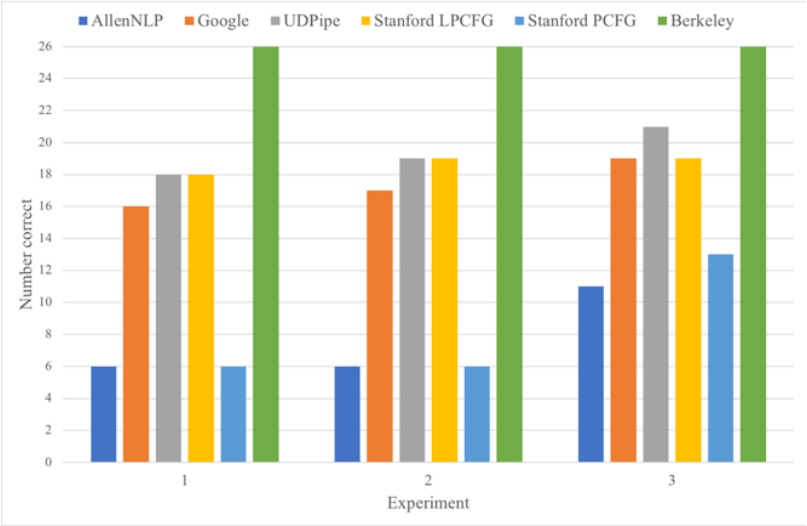


**FIGURE 14: ALL PARSERS COMPARED.**

However, the real surprise is the (statistically significant) perfect score attained by the Berkeley parser across all three experiments. Thus, only the Berkeley parser could claim to reproduce "*human level performance*" on this particular task. However, it would be incorrect to extrapolate this further; after all, the Berkeley F-scores on test corpora are now considered unexceptional, i.e., it is easy to construct a test in which the Berkeley parser is outperformed by its cohorts (described below).

## AN ADDITIONAL EXPERIMENT

[17] Both of these constituent-based parsers are trained on the PTB corpus.

We observe that the Berkeley parser's perfect behavior on Experiments 1–3 can be reproduced by a parser that uses the simple policy of always re-attaching a direct object of the first verb as the subject of the second verb (once revealed). With this in mind, consider the sentences involving sentential subjects in (10a–c), adapted from (Alrenga 2005: 177):[18]

(10) a. That the Giants lost/stole the World Series really sucks.
b. That the Giants lost/stole the World Series surprised me.
c. For the Giants to lose/steal the World Series would be terrible.

Although no temporary ambiguity is involved, the examples in (10a–c) share an important property with the Traxler experiments: *viz*. both involve biclausal structures in which the first verb is nonmatrix. In (10a–c), the verb *lose* or *steal* is embedded within the sentential subject, and therefore the verb at the end of the sentence must be analyzed as the matrix verb. Should the Berkeley parser employ a simple re-attachment policy, it will always (incorrectly) make *the World Series* the subject of the second verb.

Figure 15 summarizes the performance of all the parsers mentioned in this paper across the examples in (10a–c).

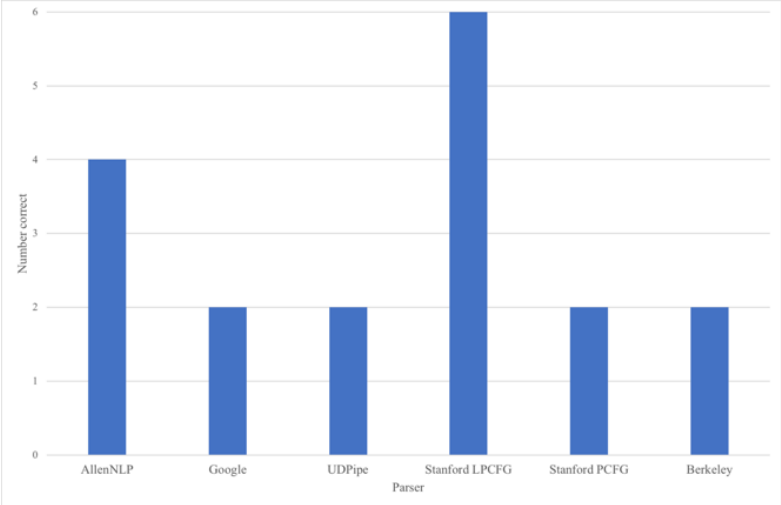


**FIGURE 15: RESULTS FOR ALL PARSERS ON SENTENTIAL SUBJECTS.**

Unlike the case of the temporary ambiguities, the Berkeley parser's performance here is unremarkable: in fact, it performs at the same level as the Google and UDPipe models. The results also demonstrate that the Berkeley parser does not have the overly simplistic re-attachment policy described earlier. In what appears to be quite a turnaround, the AllenNLP model, which performed poorly on the temporary ambiguity task, outperforms the other two other neural net models, viz. Google and UDPipe, as well as the Berkeley parser. Another surprise is that, once again, it is a legacy parser, *viz*. the Stanford LPCFG model, that has posted a perfect score.

## CONCLUSIONS

[18] In the interests of constructing a slightly larger test set, we also substituted *steal* for *lose*.

The statistical parsing enterprise has always been predicated on the idea that knowledge of language can be machine learned.[19] Unfortunately, a NN model hides what it has learned in humanly undecodable numbers. Therefore, experiments of the type described in this paper can help to shed light on this topic.[20]

It is not clear to us that the NN-based models have learned the following two very simple facts about English:

(12) A. Matrix subjects are (usually) needed.
B. Rule A. can disambiguate when verbs can be transitive or intransitive.

Rules (12A–B) are sufficient to handle all the temporary ambiguities in (1)–(3), and, following Traxler, it is likely that human subjects have incorporated them. (10a–c) also require the knowledge that clausal subjects may be introduced in English by *that* and *for*. Note sentential subjects are attested in statistical training data. Examples (13a–e) are excerpted from the PTB WSJ dataset.

(13) a. That Moscow […] would try to sell high technology to Japan […] sounds like a coals-to-Newcastle notion.
b. That they retain Korean citizenship and ties is a reflection of history […]
c. That he was the A's winningest pitcher […] indicates he may have some evening up coming […]
d. For the U.S. to lend even the slightest support to the most infamous killers on Indochina's bleak scene could only disturb America's allies elsewhere.
e. For Mr. Kageyama to argue that American employees must passively accept a direct imposition of the Japanese way of doing things is outright cultural chauvinism of the first order.

Finally, we believe our results indicate that both the discrete infinity property of human language and adversarial testing should be taken seriously. Stepping outside of the *n*-fold validation paradigm, the "all over the map" results suggest that relative performance of the statistical models can be rather difficult to predict.

## REFERENCES


Alrenga, P. 2005. A Sentential Subject Asymmetry in English and Its Implications for Complement Selection. *Syntax* 8:3, 175–107.

Andor, D., C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov & M. Collins. 2016. Globally Normalized Transition-Based Neural Networks. https://doi.org/10.18653/v1/P16-1231.

Chen, D. & C. Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, 740–750. https://doi.org/ 10.3115/v1/D14-1082.

Cheng, M., J. Yi, H., Zhang, P-Y. Chen & C-J Hsieh. 2018. Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples. arXiv:1803.01128.

Dozat, T. & C. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. *ICLP 2017 conference paper. cite arxiv:1611.01734.*


[19] It is interesting to note that Google's parser has access to more training data than the legacy constituent-based parsers. In addition to a dependency version of the PTB, it has also been trained on the Treebank Union (OntoNotes corpus/English Web Treebank/Question Treebank). Additionally, "silver" Treebanks can be added to the mix for semi-supervised training.

[20] Unfortunately, these experiments do not point to a fix. The following Stanford CoreNLP FAQ makes this situation clear:
I don't [understand/like/agree with] the parse tree that is assigned to my sentence. Can you [explain/fix] it? "*While our goal is to improve the parser when we can, we can't fix individual examples. The parser is just choosing the highest probability analysis according to its grammar*."

Dyer, C., A. Kuncoro, M. Ballesteros, N. Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 199–209. https://doi.org/ 10.18653/v1/N16-1024.

Google Has Open Sourced SyntaxNet, Its AI for Understanding Language. *Wired Magazine*. May 2016. https://www.wired.com/2016/05/google-open-sourced-syntaxnet-ai-natural-language.

Humboldt, von, W. 1836 (1999). *On Language: On the Diversity of Human Language Construction and its Influence on the Mental Development of the Human Species*. M. Losonsky (Ed.) Cambridge University Press.

Klein, D. & C. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 423–430. http://aclweb.org/anthology/P03-1054.

Kong, L., C. Alberti, D. Andor, I. Bogatyy & D. Weiss. 2017. DRAGNN: A Transition-based Framework for Dynamically Connected Neural Networks. *cite arxiv:1703.04474*.

Kong, L., A. Rush & N. Smith. 2015. Transforming Dependencies into Phrase Structures. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, 788–798. https://doi.org/ 10.3115/v1/N15-1080.

Kummerfeld, J., D. Hall, J. Curran & D. Klein. 2012. Parser Showdown at the Wall Street Corral: An Empirical Investigation of Error Types in Parser Output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1048–1059. http://aclweb.org/anthology/D12-1096.

Kuncoro, A., M. Ballesteros, L. Kong, C. Dyer, G. Neubig & N. Smith. 2017. What Do Recurrent Neural Network Grammars Learn About Syntax? In *Proceedings of the 15th Conference of the European Chapter of the ACL*: Vol. 1, 1249–1258. http://aclweb.org/anthology/E17-1117.

Kurakin, A., I. Goodfellow & S. Bengio. 2016. Adversarial Examples in the Physical World. arXiv:1607.02533.

Marcus, M., M. Marcinkiewicz, & B. Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19, 313–330. http://aclweb.org/anthology/J93-2004.

McCoy, R.T., R. Frank & T. Linzen. 2018. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks. arXiv:1802.09091v3.

Nivre, J. 2006. *Inductive Dependency Parsing*. Vol 43, Springer-Verlag New York, Inc.

Nivre J. 2015. Towards a Universal Grammar for Natural Language Processing. In *Computational Linguistics and Intelligent Text Processing*, Gelbukh A. (ed), CICLing 2015. Lecture Notes in Computer Science, v9041. Springer.

Petrov, S. & D. Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Proceedings of HLT-NAACL*. http://aclweb.org/anthology/N07-1051.

Petrov, S. 2016. Announcing SyntaxNet: The World's Most Accurate Parser Goes Open Source. *Google AI blog*. May 12th 2016. https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html.

Straka, M. & J. Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 88–99. https://doi.org/ 10.18653/v1/K17-3009.

Traxler, M, 2002. Plausibility and subcategorization preference in children's processing of temporarily ambiguous sentences: Evidence from self-paced reading. The Quarterly Journal of Experimental Psychology, 55A (1), 75–96.

Wilcox, E., R. Levy, T. Morita & R. Futrell. 2018. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 211–221.